VISION BASED METRIC-TOPOLOGICAL LOCALIZATION FOR UGV

YANG SHUAI

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirements for the degree of
Doctor of Philosophy

2017

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

....

Date YANG SHUAI

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Prof. Wang Han for his invaluable guidance and continuous support throughout my Ph.D study and research. His instruction helped me in all the time of research and writing of this thesis. Without his support, this thesis could not have reached its present form. To me, he is not only an advisor on research, but also a mentor on life.

I wish to express my thanks to all the friends and fellow labmates in Autonomous Robotics Research Lab. I wish to thank Dr. Zhou Lubing and Dr. Mou Wei for their kindly help and instruction in my research. To Mou Xiaozheng, Zhao Wei, Yuan Shenghai, Soner Ulun, Zhang Handuo and Xi Tao for their suggestions and generous help. To Dr. Guo Fanghong, Jiang Wentao, Yue Yufeng and Wang Yuanzhe for their friendship and supports during my research. I am also thankful to Jiang Rui from National University of Singapore for his assistance and corporation in the completion of the URBAN-NAV project. Working with the kind and talented fellow researchers has been a rewarding experience.

Last but not least, I would like to express my sincere thanks to my family for their constant love, encouragement and support. Without them as my backup, I will not be where I am today.

Abstract

This thesis studies vision based localization methods for unmanned ground vehicle (UGV) to achieve accurate and robust positioning in GPS challenging environments. Efforts are made from the perspective of topological and metric localization. Due to the incremental nature, visual odometry belongs to metric localization category. For a monocular visual odometry system, drift and scale ambiguity are the main issues that restrict it from extensive applications in autonomous navigation. In this thesis, a metric localization approach based on the fusion of visual odometry and road constraints is proposed. The drift and scale ambiguity of monocular visual odomtry are both considered as measurement uncertainties and incorporated into a presented Gaussian-Gaussian Cloud model. The geometric shapes of road networks are considered as constraints to assist with position estimation. Shape matching method is utilized to evaluate the alignment between historical trajectory from visual odometry and road shape from digital map.

As a typical topological localization approach, place recognition is playing important roles in mobile vehicle navigation. Most of the current place recognition methods are designed for the application in a particular environment (e.g. indoor or urban environment). In this thesis, a place recognition method which is applicable to various environments is presented. A modified vocabulary tree with the ability of merging multiple kinds of features is designed to customize different combination of features for different environments.

The downsides of pure place recognition and road-constrained metric localization are obvious. Place recognition approach suffers from its discontinuous output, while road constrained metric localization suffers from the on-road assumption as well as the tough initialization. To play their respective advantages, a metric-topological localization approach based on the integration of place recognition, visual odometry and road constraints is proposed. Topological and metric modules run in a parallel way and a mutual check scheme is utilized to ensure the consistency of the positioning results.

When information sources from other sensors are available, a proper sensor fusion technique is required. To this end, a fusion approach is proposed to localize vehicles by integrating a visual odometry, a low-cost GPS, and a two-dimensional digital road map in this thesis. The concept of artificial potential field that is widely used for obstacle avoidance is leveraged to represent measurements and constraints, respectively. Position measurements from visual odometry and low-cost GPS are modelled with a potential well function, while road constraints from digital map are modelled with a potential trench function without additional map matching. By searching for the minimum of the combined potential field, the position can be estimated. All the approaches developed in this thesis are extensively and successfully verified on real world datasets.

List of Abbreviations

BA	bundle adjustment
BRIEF	binary robust independent elementary features
BOW	bag of word
CNN	convolutional neural network
DIRD	illumination robust descriptor
DOF	degree of freedom
EKF	extended Kalman filter
FAST	features from accelerated segment test
GGC	Gaussian-Gaussian Cloud
GGD	Gaussian-Gaussian Distribution
UGD	Uniform-Gaussian Distribution
GPS	global positioning system
IMU	inertial measurement unit
KF	Kalman filter
MM	map matching
MVO	monocular visual odometry
NTU	Nanyang Technological University
ORB	oriented FAST and rotated BRIEF
OSM	OpenStreetMap
RANSAC	random sample consensus
SIFT	scale-invariant feature transform
SLAM	simultaneous localization and mapping

SURF	speeded up robust feature
UGV	unmanned ground vehicle
VO	visual odometry
UV	unmanned vehicle
2D	two-dimensional
3D	three-dimensional

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Overview

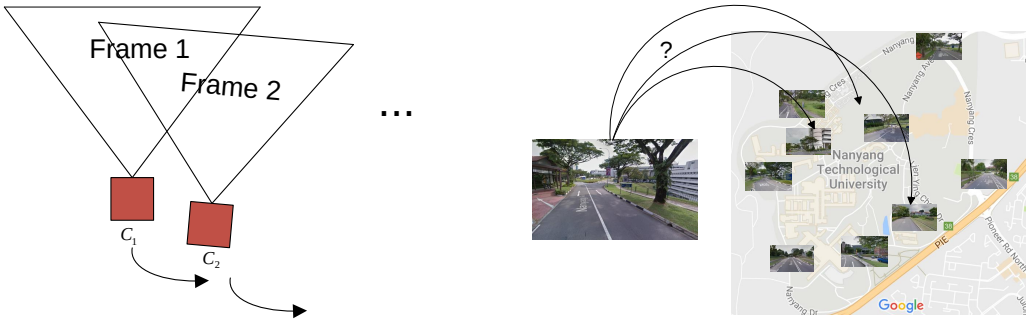
In the past decade, with the progress of artificial intelligence, unmanned vehicles (UVs) are taking a growing role in the development of modern society. As shown in Fig. 1.1, logistic companies are using robots to do warehouse management and transportation; more and more car manufactures or even start-ups have launched their self-driving programs; service robots such as restaurant robot waiter and hotel room service robot are performing jobs that is dirty, dull or repetitive. Their good properties such as self-replication ability, upgradable and deep learning capability make intelligent robots involved in virtually every sector of our economy. Manpower is released and we human beings are stepping into a new period.

One of the most basic requirement for unmanned vehicle is its self-localization ability. Whatever a self-driving car, a bomb disposal robot, or even a self-guided vacuum cleaner, the position has to be obtained during task. Most commonly used methods for self-localization are sensor-based, which use sensory information to locate the robot in its environment. Many different sensors have been used, among which the Global Positioning System (GPS) is the most popular for outdoor environment. GPS

Figure 1.1: Application of unmanned vehicles. The three figures from left to right show logistic robots, driverless car and room service robot.

can provide a good accuracy under ideal condition; however, it can be inaccurate or unavailable due to obstacles such as skyscrapers and trees. GPS signals can also be affected by multi-path issues [1], which tend to cause significant errors on robot localization. 3D sensors like scanning laser range finders are also widely used (e.g., the Google car [2]) due to their high accuracy (e.g., 1 mm, phase range finders), low energy consumption and high reliability in complex environments. However, such sensors can cost tens of thousands of dollars. Other sensors such as inertial measurement unit (IMU) [3] have also been used for localization purpose. But they generally lack the ability of environment perception.

Recent years, navigation based on visual sensors has drawn great attention due to its low cost and possibility to provide a full 6 degree of freedom (DOF) motion estimation. Generally, there are two kinds of vision-based localization approaches according to the continuity of the positioning results. The first category is called metric localization due to the accumulative nature. Visual odometry (VO) is one of the most representative approach of metric localization. It estimates the instantaneous motion using the input from a camera rigidly attached to the body of the mobile agent. As the agent moves, the ego-motion is estimated recursively as depicted in Fig. 1.2a. Thus, a continuous trajectory can be estimated. However, with this kind of approach, the small errors of motion estimation accumulate over time and result in drift on localization. Visual simultaneous localization and mapping (SLAM) is an improved VO. Instead of estimating ego-motion alone, the map of the environment is created simultaneously. One important feature of visual SLAM is its loop closure detection module, which allows to correct accumulated drift after large loop closures. That is, when a place has been revisited, the accumulated error can



(a) Visual odometry. (b) Place recognition.

Figure 1.2: Representative vision-based metric and topological localization approach, separately.

be corrected through the position constraints given by visual loop-closure detection. Hence, the ability of recognizing a pre-visited place or place recognition ability is critical for visual SLAM.

The second category of vision-based localization is called topological localization, which refers to the identification of the discrete location of the mobile agent. Instead of finding the metric position, only a rough position information, such as "I am in the office", is given in topological case. As shown in Fig. 1.2b, place recognition can be interpreted as: given one query image taken at anywhere of the world, find one or more images in the geo-tagged database depicting the same place. The matching process between the query image and the database can be considered as an image retrieval scheme in computer vision field. Since the image database is generated in discrete way, place recognition belongs to topological category. The main challenge place recognition faced is no longer the drift issue but how to cope with appearance variations and perceptual aliasing. In addition to loop closure in SLAM, place recognition can be very effective in scene understanding and semantic mapping, as well as map fusion. Incorrect place recognition can cause large errors in a SLAM or map fusion context, or break down a scene interpretation system, emphasizing the importance of robust place recognition.

1.2 Objectives and Contributions of This Study

In this thesis, we aim to develop robust vision based localization methods for unmanned ground vehicle in GPS challenging environments. Efforts are made from the perspective of metric and topological localization. Firstly, a road-constrained metric localization approach based on VO is proposed. Then a topological localization approach based on place recognition is developed. To play their respective advantages, an integrated approach which combines topological and metric localization is also explored. Finally, to deal with situations where multiple information sources are available, a sensor fusion approach is presented to localize vehicles. All the approaches developed in this study are extensively and successfully verified on real world datasets.

Specifically, the main contributions of this thesis are as follows:

1. **A metric localization approach based on the fusion of VO and road constraints (Chapter 3).**

   Studies on monocular VO have been very popular in the past decade, and localization approaches using one single camera are very desired. Nevertheless, there are two main challenges: drift and scale ambiguity, restricting monocular odometry from an extensive application on real autonomous navigation. Inspired by the concept of cloud, a new Gaussian-Gaussian Cloud model is proposed to give a unified representation of the measurement randomness and scale ambiguity in monocular VO. In this model, a collection of cloud drops is generated. Both the drift and scale ambiguity are considered and represented simultaneously in each cloud drop. To reduce the measurement uncertainties of any drop in Gaussian-Gaussian Cloud, road constraints from the geometric shapes of road network from the open source map—OpenStreetMap are utilized. The map is firstly converted to a template edge map and a shape matching step is then implemented to assign probability of each cloud drop,

indicating what degree the drop accords with road constraints. A parameter estimation scheme is used to narrow down the scale ambiguity of monocular VO while resampling cloud drops. Evaluations on the KITTI benchmark dataset and our self-collected dataset have demonstrated the stability and accuracy of the proposed approach.

2. **A topological localization approach based on place recognition (Chapter 4).**

   Place recognition has been intensively studied in the context of robot vision and bag-of-word (BOW) based approach gains its popularity for its efficiency and robustness. No matter what kind of place recognition system, an appropriate appearance representation method should be chosen. Many image features have been examined in the past for place recognition purpose. However, there is no such feature that outperforms others in all environments. Each feature has its own advantage, thus, they should be carefully chosen depending on the contexts and environments. In this thesis, the author proposes a place recognition method which is applicable to various environments. The core of this method is a modified vocabulary tree that has the ability of merging multiple kinds of features. With this design, users can customize different combination of features for different environments. Note that "modified" here is relative to traditional BOW vocabulary tree. The system has been tested in real-time on real-world datasets and the experiment results demonstrate the advantage of our system compared with existing approaches.

3. **A metric-topological localization approach based on the integration of place recognition, VO and road constraints (Chapter 5).**

   Visual odometry, road constrained methods and place recognition are all popular approaches to localize a mobile vehicle from three different perspectives. Separate implementation of these methods may cause the localization system vulnerable due to the drift issue and local pose estimation of VO, the on-road assumption and tough initialization of road constrained methods, and the dis-

continuous output of place recognition. In order to give full play to their advantages, an integrated localization strategy is presented in this study, where the metric information from VO measurement as well as digital map, and the topological information from place recognition are incorporated. Place recognition assists initialization process and provides topological place estimation at all times. Gaussian-Gaussian Distribution is used for VO raw measurement representation such that the errors of odometry is appropriately modelled. By comparing similarities between the digital map and odometry trajectories, we then use road constrained approach to correct odometry estimation. Finally, a mutual check gives a criterion for judging whether metric and topological results are sufficiently consistent. Experiment results show that the integrated system outperforms subsystems with mean localization error at 2.9 meters on our self-collected dataset with off-road scenarios.

4. **A metric localization approach based on the fusion of VO, low-cost GPS and digital map (Chapter 6).**

   In the previous studies, we focus on the development of vision-based localization without the participation of other sensors. When other information sources are available, a proper sensor fusion technique is desired. To this end, we present a fusion approach to localize vehicles by integrating a VO, a low-cost GPS, and a two-dimensional digital road map. Distinguished from conventional sensor fusion methods, two types of potential functions (i.e. potential wells and potential trenches) are proposed to represent measurements and constraints, respectively. By choosing different potential functions according to data properties, data from various sensors can be integrated with intuitive understanding, while no extra map matching is required. The minimum of the fused potential is regarded as final position estimation. Experiments under realistic conditions have been conducted to validate the satisfactory positioning accuracy and robustness compared to pure VO and map matching methods.

1.3 Outline of This Thesis

The remainder of this thesis is organized as follows:

Chapter 2 gives an extensive survey of visual based localization methods. The main approaches published in the last ten yeas with regard to our common thread are reviewed.

Chapter 3 details the metric localization approach which is based on the fusion of VO and road constraints. The goal of this approach is to globally localize a mobile vehicle equipped with a single camera and a freely available digital map. Gaussian-Gaussian Cloud model which gives a unified representation of the measurement randomness and scale ambiguity in monocular VO is interpreted. Shape matching process which incorporates road constraints from the geometric shapes of road network into the localization framework is explicated. Evaluation details on the KITTI benchmark dataset and our self-collected dataset are given.

Chapter 4 elaborates the proposed place recognition method which is applicable to various environments. The reasons why such a place recognition method is desired are explained first. Then the framework for combining point and line features is introduced. Finally, the experiments conducted on real-world datasets are presented.

Chapter 5 explains the metric-topological localization approach which is based on the integration of place recognition, VO and digital map. Topological localization based on place recognition and metric localization based on VO and digital map are presented. Their pros and cons are discussed. The strategy which integrates topological and metric localization is introduced. Evaluation details on our self-collected dataset are given.

Chapter 6 introduces the metric localization approach based on the fusion of VO, low-cost GPS and digital map. The concept of potentials is explained first. Then the two types of potential functions (i.e. potential wells and potential trenches)

which are used to represent measurements from VO as well as low-cost GPS and constraints from digital map are introduced. Experiment results under realistic conditions are given.

Chapter 7 summaries this thesis. The contributions of this work together with the expected future work are provided.

Chapter 2

Technical Background and Literature Review

As a replacement or enhancement of GPS, vision-based positioning problem has a long history and many solutions have been developed. Among these solutions, visual odometry, visual SLAM and place recognition based approaches are the majority. Approaches which combine vision-based positioning and other information sources account for most of the remaining. In this chapter, an extensive survey of the related technology and literature is provided. The main approaches published in the last ten yeas with regard to vision-based localization method are reviewed.

The remaining part of the chapter proceeds as below. In Section 2.1, we explain how visual odometry problem is formulated and discuss the related work according to the sensor configuration. The problem of visual SLAM is introduced in Section 2.2 and the corresponding literatures are given. Section 2.3 describes map-assisted localization approaches, where online maps are utilized to assist vehicle localization. Place recognition problem and the related works are depicted in Section 2.4. Section 2.5 concludes the chapter.

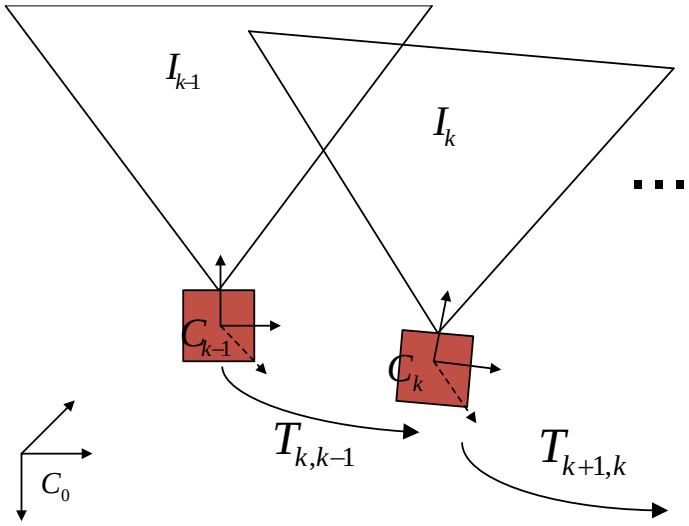


Figure 2.1: An illustration of visual odometry positioning. The absolute camera pose $\mathbf{C}_k$ with respect to the initial coordinate frame $\mathbf{C}_0$ is computed from the concatenation of instantaneous transformations $\mathbf{T}_{k,k-1}$ between the adjacent cameras, which are computed from visual features.

2.1 Visual Odometry

Fig. 2.1 depicts how the camera pose is computed in visual odometry problem. $\mathbf{C}_0$ is the pose of initial coordinate frame. $\mathbf{I}_{k-1}$ and $\mathbf{I}_k$ are the images of consecutive frames with camera pose $\mathbf{C}_{k-1}$ and $\mathbf{C}_k$ respectively. The relationship between $\mathbf{C}_{k-1}$ and $\mathbf{C}_k$ can be expressed as [4]:

$$\mathbf{C}_k = \mathbf{C}_{k-1}\mathbf{T}_{k,k-1}, \tag{2.1}$$

where $\mathbf{T}_{k,k-1} \in \mathbb{R}^{4\times4}$ is the rigid body transformation matrix between time instant k and $k-1$. $\mathbf{T}_{k,k-1}$ can be further expressed as:

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2.2}$$

where $\mathbf{R}_{k,k-1} \in SO(3)$ and $\mathbf{t}_{k,k-1} \in \mathbb{R}^3$ denote the rotation and translation parts of $\mathbf{T}_{k,k-1}$, respectively.

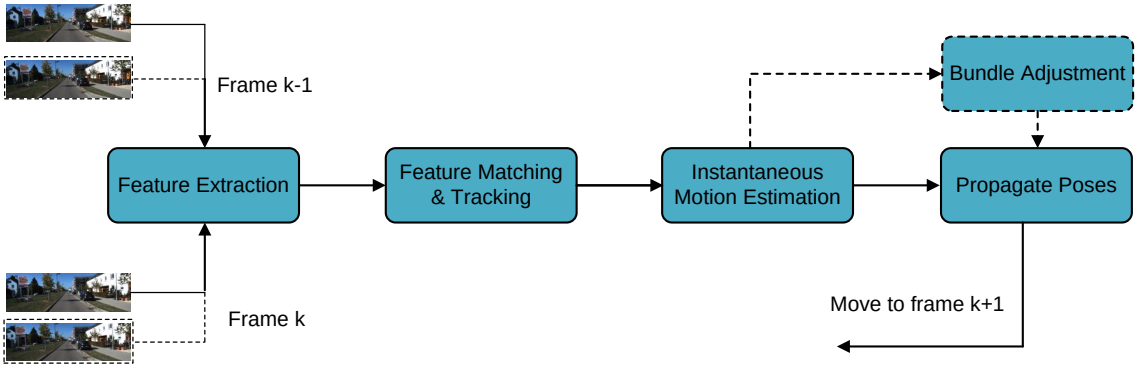


Figure 2.2: The pipeline of visual odometry system.

The purpose of visual odometry is nothing but computing $\mathbf{T}_{k,k-1} \in \mathbb{R}^{4\times4}$ at every time step k as accurate as possible. Then the current camera pose is computed by concatenating all of the instantaneous transformations. The set of all the camera poses $\mathbf{C}_{0:k} = \{\mathbf{C}_0, \mathbf{C}_1, ..., \mathbf{C}_k\}$ forms the camera's trajectory. Since the estimated instantaneous transformation $\mathbf{T}_{k,k-1}$ can not be completely accurate; small errors from $\mathbf{T}_{k,k-1}$ accumulate over time, which causes a drift of the estimated camera pose. The drift issue is one of the biggest challenge faced by visual odometry.

In 2004, the first complete visual odometry system, which operates in real time was proposed by David Nister [5]. In their system, a point feature tracker scheme is designed to deal with feature correspondences between consecutive frames, and after which, a robust motion estimation step is implemented to compute motion from visual input alone. Inspired by this work, more and more attentions [6], [7], [8] are given to visual odometry and the modern structure of visual odometry as illustrated in Fig. 2.2 begins to take shape. Visual odometry has played very important roles in many kinds of mobile robotic systems, such as ground vehicles [9], unmanned aerial vehicles [10] and underwater autonomous vehicles [11]. But the most well known application must be the Mars exploration program [12], where visual odometry is developed to assist Mars Exploration Rovers (MER) to get their position and the good positioning results demonstrate the capability of visual odometry.

From the perspective of sensor configurations, visual odometry can be classified

into three categories, namely monocular visual odometry, stereo visual odometry and RGB-D visual odometry. For monocular visual odometry, only one camera is used, which makes monocular odometry very flexible and cost-effective. But its disadvantages are equally notable. The motions estimated are up to an unknown scale [13] due to the unawareness of depth. Efforts need to be made to obtain this unknown scale. For stereo visual odometry, a calibrated stereo rig which provides the geometry constraints is used. Compared with monocular ones, the scale ambiguity does not exist any more and it is more widely used due to its good applicability. But stereo rig usually costs much and occupies a larger space. Besides, the calibration process is cumbersome and it is difficult to get an accurate and durable calibration [14]. For RGB-D case, one Microsoft Kinect or the Asus Xtion camera which has the depth sensing capability is used. It has the advantages of both stereo and monocular approaches. However, it can be used indoors only thanks to its limited ranging distance [15]. All the three approaches have their own characteristics, but they all aim at providing precise localization with low consumption. The two-part tutorial series [4], [16] published in 2011 and 2012 give a very detailed survey on the progress of visual odometry. In the following sections, we mainly focus on works published after that.

2.1.1 Monocular Visual Odometry

Since only one camera is involved in monocular vision systems, no camera's calibration or synchronization is required. Moreover, monocular camera covers a wider filed of view than stereo one. It has great potentials on robot navigation and autonomous driving. But monocular approaches suffer from the scale ambiguity of the translational estimation. To make monocular algorithms applicable to localization purpose, several solutions have been proposed.

The most straight forward approach is to combine information from IMU, GPS or wheel encoders [17], [18], which can provide scale reference, with vision system. The

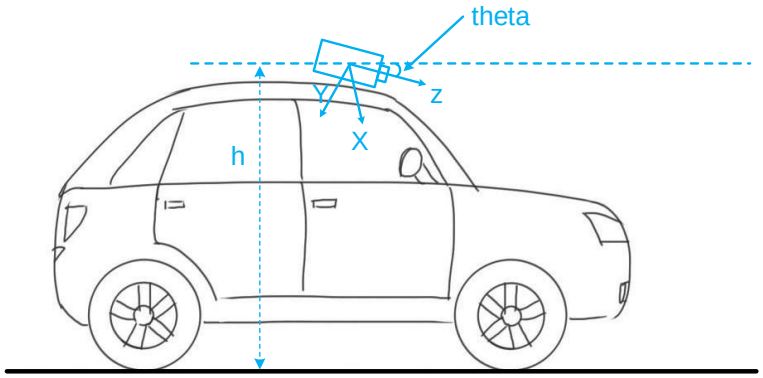


Figure 2.3: Camera configuration of ground plane aided scale correction approach. h and θ denote the camera height above the ground and the pitch angle, receptively.

authors of [18] propose a depth enhanced monocular odometry, where depth information provided by RGB-D camera or lidar is associated with cameras. Features whose depth determined by triangulation using the previously estimated motion of the camera are added to the point cloud to tackle the sparsity issue of depth information. Frame to frame motion estimates are refined by bundle adjustment. Even though this approach was proposed in 2014, they are still one of the best approaches in the ranking of KITTI visual odometry benchmark [1]. The combination of vision and other sensors is a viable option which of course will increase the cost and complexity of the system.

Another popular method to estimate the metric scale is assuming that the camera is moving at a known and fixed height over a plane ground [19], [20], [21]. The result of this kind of method relies heavily on the accuracy of ground plane detection. Fig. 2.3 illustrates how the camera is mounted on the vehicle. h is the pre-measured height from camera centre to ground plane; θ is the non-zero pitch angle; p is a point on the ground plane, with coordinate (x, y, z) in camera coordinate system. For each p,

$$h^* = y\cos\theta + z\sin\theta, \tag{2.3}$$

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

where h^* is the estimated camera height. The metric scale can be retrieved as:

$$s = \frac{h^*}{h}. \tag{2.4}$$

Based on this idea, authors of [20] develop a monocular system which is capable of correcting scale drift. In their system, multiple information sources such as sparse features and dense stereo between consecutive frames are used to determine the ground plane. In order to improve the stability in homograph decomposition, authors of [21] propose to separate motion parameters in the homograph H from structure parameters of the ground plane. An online self-supervised approach is proposed in [22], where a ground surface classier is designed. Appearance information such as ground colors is used to learn the classier and each pixel on the image is assigned with ground probability.

Some other researchers propose to use objects with known size to give the absolute scale of monocular results [23]. Nevertheless, it is quite tough to ensure that the object appears and be detected in all frames. When extra information sources are not available, authors of [24], [25], [26] presented an alternative approach, where the translation norm between the first two cameras is configured to 1 and the following camera poses are estimated with respect to the first two poses. An initialization process is always involved in these frameworks. The real motion of the whole system can be retrieved once the metric transformation between the first two frames is figured out.

2.1.2 Stereo Visual Odometry

Compared with monocular vision systems, stereo ones have additional spatial constraints which provide 3D sensing capability. The estimated motion as well as the 3D structure of the scene is at the real scale. Localization can be well achieved without the assistance of other sensors in unknown environments [27]. 3D point

cloud generated from disparity image can be incorporated into scene understanding or object avoidance threads [28]. Though it is more troublesome to configure multi cameras, stereo vision systems are still the most widely used solution in real application. Maximizing the benefits of binocular is one of the effort directions of current works. Here, we review the related ones according to the structure as illustrated in Fig. 2.2.

Feature extraction is the very first and time consuming step, which includes feature detection and description. Very detailed comparisons between different detectors and descriptors are given in [16]. In nature scenes, features usually locate on salient objects, which generates over-centralized feature distribution. To overcome this, a bucketing approach is leveraged in [11], [29], where the input images are divided into n buckets and each bucket has no less than m features. As a result, at least $n * m$ features are distributed evenly across each input image. In order to deal with inaccurate image rectification, a reweighting scheme is employed in [30], where lower weights are given to features that are far way from image centre. For real time application, robust and computation cost-efficient descriptors are very desired. Authors of [31] present a visual odometry method that uses a newly designed descriptor to achieve good feature matching. Binary features such as ORB are getting wider usage [32] due to their faster speed in both computation and matching. A parallel process of left and right frames also increases feature extraction speed.

Feature matching step is responsible for finding the corresponding features between left and right, as well as previous and current images. Feature descriptor distances (e.g. Euclidean and Hamming distance) are commonly used to determine the feature similarity [33]. Mutual best matching check scheme [16] is a trick to ensure that the correspondences are the best in both left to right and right to left orders. In some implementations, the feature matching process forms a loop [29], which goes as: given current and previous image pairs, for each feature u_k^l detected in current left image, try to find the corresponding feature u_k^r in the current right image first; if u_k^r is found, try to find the corresponding feature u_{k-1}^r in the previous right image; if

u^r_{k-1} is found, try to find the corresponding feature u^l_{k-1} in the previous left image; once the four correspondences are found, they will be fed into motion estimation step, otherwise they will be abandoned. By using this loop, both left to right and previous to current consistencies are kept. Instead of only using consecutive frames, authors of [34] propose a novel stereo visual odometry approach which uses the whole history of the tracked points to compute the motion of the camera and very impressive results are achieved.

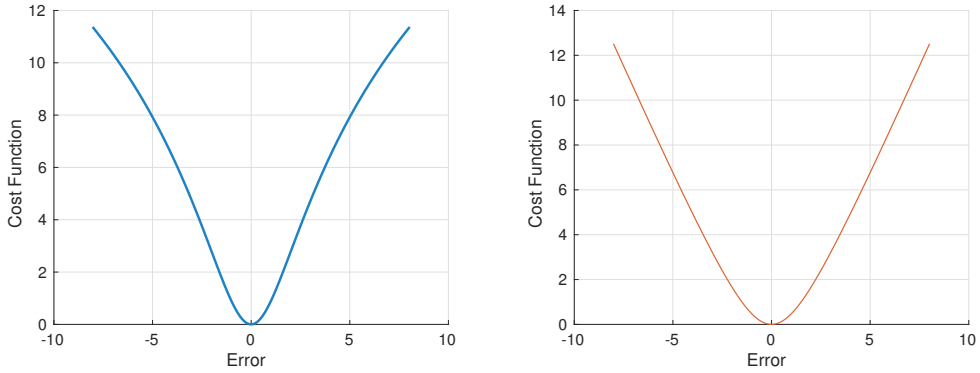
The successfully tracked features from feature matching will be fed into motion estimation step, which is frequently completed through a reprojection error minimization process [16]. The cost function is defined as:

$$\underset{\mathbf{T}_{k,k-1}}{\operatorname{argmin}} \sum_i ||\mathbf{u}^i_k - \mathbf{u}^i_{k-1}||^2, \tag{2.5}$$

where $\mathbf{u}^i_k$ is the pixel position of ith feature in current frame and $\mathbf{u}^i_{k-1}$ is the pixel position of the same feature in previous frame. Gauss-Newton [35] and Levenberg-Marquardt [36] algorithms are frequently used to solve this non-linear least square problem.

Due to image noise, inaccurate camera calibration and illumination changes, the tracked features consist of wrong data associations. The feature points are clustered into two categories, namely inliers and outliers. An outlier rejection scheme is required to filter out feature points that are inappropriate for reprojection error minimization. As a typical model fitting approach in the presence of outliers, random sample consensus (RANSAC) has been widely used in visual odometry [29]. It works fine when there are only a small percentage of outliers. But when outliers take a larger proportion, the required iteration numbers increase exponentially, which causes great difficulties in real time implementation. Recent years, many variants of RANSAC have been proposed to overcome the above drawbacks [37].

Another alternative to filter out outliers is non-iterative method [38], [39], which has



(a) Cauchy cost function. (b) Huber cost function.

Figure 2.4: Robust cost functions used in outlier detection.

a significant reduction in computation. By assuming the reprojection errors follow Gaussian distribution, the effects of outliers are eliminated through a robust cost function due to their incompatibility with Gaussian distribution. Graphs of some cost functions (Cauchy and Huber) are depicted in Fig. 2.4. Note that both of the cost functions are convex. Besides, when the errors are at small values, the cost functions forms near quadratic; when the errors are larger than a threshold, the cost functions become linear. Based on the robust cost function, outlier removal approach which is suited for high-speed scenario is presented in [40]. They find that simply based on fixed threshold to detect outliers is improper for environments with large-scale depth. In their work, inliers and outliers are separated with an iterative alternating scheme. Authors of [41] present a statistic-based outlier removal technique, which works better when there are large overlaps between consecutive frames.

With all the remaining inliers, a 6 DOF transformation can be computed by minimizing the reprejection error between consecutive frames. Nevertheless, without considering the longer consistency between multiple frames, errors originated from local minimum or wrong matching propagate. Thus, a local optimization or windowed bundle adjustment scheme is desired so that additional constraints from neighbour frames are taken into account [42], [43]. A pre-condition for local minimization is to find successfully tracked features over more than two frames. Normally, the cost

function is defined as:

$$\underset{\mathbf{C}_k,\mathbf{X}^i}{\operatorname{argmin}} \sum_{i,k} ||\mathbf{u}^i{}_k - \pi(\mathbf{X}^i, \mathbf{C}_k)||^2, \tag{2.6}$$

where $\mathbf{X}^i$ is the 3D coordinate of the ith landmark; $\mathbf{u}^i{}_k$ is the pixel position of $\mathbf{X}^i$ in current frame; $\pi(\mathbf{X}^i, \mathbf{C}_k)$ is a warp function that projects $\mathbf{X}^i$ to camera coordinate according to the reference frame $\mathbf{C}_k$. Both the landmark 3D positions as well as camera states are optimized through this cost function. The tricks on how to trade off the computation cost and robustness are discussed in [16].

2.1.3 RGB-D Visual Odometry

RGB-D camera is a sort of low-cost sensor which captures RGB color image together with a corresponding depth image. Compared with conventional monocular, RGB-D's depth sensing ability is its greatest advantage. Compared with stereo, the depth image obtained from RGB-D is more dense since every pixel is augmented with depth values. All these strengths make RGB-D sensor an popular option for robotics applications [44], [45]. Specific to localization purpose, RGB-D sensor has a great application prospect especially for indoor environment. Recent years, more and more attentions are being paid to RGB-D visual odometry.

Due to the mixture nature of RGB-D sensors, techniques developed for monocular and stereo visual odometry can be easily applied on RGB-D odometry. Following the framework of conventional feature based visual odometry, authors of [46], [47] present a real-time RGB-D odometry system for unmanned aerial vehicle (UAV). In their system, special focuses are given to the analysis of depth uncertainty. In [48], iterative closest point (ICP) is leveraged to assist RANSAC to find the best alignment between frames and constraints from ICP are incorporated into sparse bundle adjustment (BA). Very dense 3D modelling of indoor environments is achieved in their work.

Indoor environment tends to be well structured, where high level features (e.g. line and plane) are easily detected. This is a very good property because point features are more sensitive to illumination variations than line features, while line features have more position ambiguities than point features. Besides, it is more difficult to find point feature correspondences than plane feature correspondences in the presence of wider displacements between consecutive frames. Thus, a proper combination of point and line features or point and plane features seems to improve robustness and accuracy of visual odometry. Authors of [49] present a robust RGB-D odometry framework, in which both point and line features are considered and reprojection errors from both of them are fused to do motion estimation. Conclusions are made that the motion estimation from the fused approach have a smaller uncertainty than results from each feature alone. Authors of [50] present a similar odometry system, where point features are used only when not enough plane correspondences are found. Experimental results demonstrate that plane features have a better performance in wider baseline situation.

Though odometry systems based on line or plane features utilize more image information than systems based on point features, there are still a lot of image pixels wasted. Additionally, it's quite difficult to find enough feature correspondences in low texture scenes. Feature based odometry could become unworkable in such conditions. Finally, detecting and describing features are time consuming, no matter it's point or line features. In recent five years, direct methods that do not have the above drawbacks are getting the second wind [25], [51]. Instead of relying on limited number of features, direct methods use information from all image pixels or selected pixels to estimate relative motion. With more image pixels being considered, the scenes can be reconstructed in a more dense way and complex environments like low-texture scenes are no-longer challenging. Commonly, a photo-consistency assumption goes as follows: pixels that belong to the same landmark have identical intensity. Following this assumption, image alignment is achieved through a photo-

metric error minimization process which can be defined as:

$$\underset{\mathbf{T}_{k,k-1}}{\operatorname{argmin}} \sum_{i} ||\mathbf{I}_k(\pi(\mathbf{u}^i, \mathbf{d}^i, \mathbf{T}_{k,k-1})) - \mathbf{I}_{k-1}||^2, \tag{2.7}$$

where $\mathbf{I}_k$ and $\mathbf{I}_{k-1}$ are the current and reference frame, respectively; $\mathbf{u}^i$ and $\mathbf{d}^i$ are the pixel position and depth of the ith pixel; $\pi(\mathbf{u}^i, \mathbf{d}^i, \mathbf{T}_{k,k-1}))$ is a warping function that maps $\mathbf{u}^i$ to the reference frame $\mathbf{I}_{k-1}$ through motion $\mathbf{T}_{k,k-1}$. A detailed explanation on how to form and solve this function is given in [52]. Based on the direct idea, a dense system that performs motion estimation through the minimization of both photometric and depth error is developed in [53] and very robust results are accomplished in low texture scenes. A semi-dense system which combines the efficiency of dense approach with the robustness of feature-based approach is developed in [25].

2.1.4 The Cloud Model

In a visual odometry based localization system, the modelling of measurement uncertainties from drift or scale ambiguity plays a major role. Thus, to have a better positioning performance, it is necessary to discuss measurement uncertainties. Generally speaking, randomness and fuzziness are the two most widely used terms to describe measurement uncertainties. To model them, the theory of probability and fuzzy mathematics have been respectively well explored. However, in many kinds of measurements, these two uncertainties co-exist and it is hard to distinguish them. A decade ago, one different notion called cloud model was proposed by a group of researchers in [54]. Instead of considering randomness and fuzziness separately, this model provides a unified representation of them. In Chapter 3, we use the concept of cloud model to assist our positioning. Here, a rough survey about this concept is given.

The cloud model was first proposed by Li *et al* [54],[55] for concept conversion

between qualitative and quantitative ones. In the Gaussian cloud model discussed in [54], three numerical characteristics (expected value, entropy and hyper entropy) are defined to describe a qualitative concept. By incorporating randomness and fuzziness into unified consideration, the cloud model is able to represent concepts with uncertainty, where each cloud drop contribute to the concept with a particular "degree of contribution".

The cloud model has shown promising results in modelling uncertainty. Particularly, it has been successfully implemented in situation prediction [56], path planning [57] and stochastic optimization [58]. Although the Gaussian cloud model is the most-studied cloud model as its universality, different types of clouds can be created according to different probability distributions, such as uniform cloud, power law cloud, trapezium cloud etc. The appropriate choice of cloud model is based on accurate comprehension of the physical process that describes the designated problem.

2.2 Visual SLAM

The drift issue has been preventing visual odometry from being used in long range navigation. Although the error accumulation can be reduced by local optimization, it could never be completely eliminated. Theoretically, local optimization can be extended to global. However considering computation cost, it is difficult to implement global optimization in real time application. Furthermore, the goal of visual odometry is to incrementally localize itself without generating the map of the surrounding environment. Nevertheless, understanding the environment and scene reconstruction are also critical for robot navigation. In order to achieve more robust and accurate motion estimation as well as map generation, a visual SLAM system is desired.

As the name says, SLAM system estimates the robot pose and generates the map of the environment features while moving through the unknown environment. Com-

pared with odometry, SLAM carries one additional mapping thread, which constructs and updates the map incrementally. The two threads (i.e. localization and mapping) go simultaneously and complement each other as: the estimated robot poses are exploited to improve landmark positions in the map while the estimated landmarks are exploited to improve robot poses. Based on sensor configurations, different kind of SLAMs such as WiFi-SLAM [59], laser SLAM [60] and visual SLAM have been developed and their applications cover most of the filed of robot navigation [61].

Visual SLAM has a long history. Early visual SLAM systems are originated from conventional SLAM, where sequential filtering is used [62]. Given robot control signals and feature measurements, by assuming that the state-space estimation and measurements follow Gaussian distribution, a probability framework is formulated to deal with errors from different sources (e.g. motion model and observation model). And the joint probability distribution of all the states are modelled with a Bayesian filter. To approximate the solution of Bayesian filter, Kalman Filter (KF) [63] and its variants [64], [65] are frequently used. Tricks such as Taylor Expansion are developed for non-linear estimation. Non-parametric approaches (e.g. Particle Filter) are developed for non-linear and non-Gaussian filtering. At each iteration, two steps, prediction and measurement update are concatenated to determine the state and covariance values. The main drawback of filtering approach is that the computation cost increases significantly as more and more states and features are involved in the map.

Current visual SLAM takes SLAM as graph optimization process based on bundle adjustment and loop closing technique. Local bundle adjustment discussed in Eqn. (2.6) that optimizes the structure and motion is the core of a visual SLAM framework. Double window and multi window optimization which are well-suited for different environments are also explored [66]. Instead of taking all the frames and all the features into account, the concept of "keyframe" is commonly used to maintain the sparsity of graph. A frame is selected as keyframe only if the overlap

between current frame and previous keyframe is lower than a threshold [67]. The connections between all the nodes of the graph are decreased dramatically compared with filtering approach. Real time implementation is no longer too difficult [24], [68]. A very detailed comparison study on filtering and optimization approach has been given in [69].

Loop closure detection scheme that handles the cases when the robot revisits a historical position, is another important feature in current visual SLAM framework. Successful loop closure detections (especially large scale loop closing) help the graph to keep the overall consistency and correct drifts. Loop closing is becoming the third thread apart from localization and mapping in current visual SLAM frameworks [32]. Nevertheless, loop closures do not necessarily exist in practical driving conditions. Even when loops do exist, the corrected motion is still a delayed result for the route before loop closure. Thus, loop closing method is not appropriate for applications where instantaneous decisions are desired, such as driverless car.

2.3 Map-assisted Localization

Pure odometry or SLAM based localization system could not possibly give a global position estimation without an accurate initial global location. Besides, the drift issue makes odometry an unreliable solution. Thus, other useful information sources are desired to complete the global localization goal.

With the development of Geographic Information System (GIS), information in digital maps can be utilized to assist localization. Generally, the information in digital maps can be divided into two categories: topological information and geometric information. Topological information is represented with a graph with nodes and edges, where nodes denote intersections or landmarks, and edges denote drivable roads. Google Street View image database inherently provides a topological map consists of landmark images, and it has been used for urban localization [70], [71],

[72]. In [73], map feature data is transformed to images of map features; thus the position of the vehicle can be obtained without landmark database.

Distinguished from topological information, geometric information in digital map considers metric distance and geometric shape of drivable roads. Based on the assumption that vehicles are always on a drivable road, the distance and geometric shape of roads can be regarded as constraints that correct and compensate measurement from other sensors such as odometry, GPS and inertial navigation system (INS). In [74], road constraints are expressed by curve models including but not limited to straight roads, arc roads and polynomial roads. Map matching, as one of the road constrained localization approaches, achieves error correction by matching coordinates measured by other sensors into a digital map [75], [76]. As point-by-point matching does not consider historical trajectory of the vehicle, sequential map matching has been proposed to decrease failure rate. In [77], several points are matched at once by incorporating the map matching problem into a hidden Markov model. Recently, shape matching is implemented for road constrained localization [72]. The shape matching algorithms calculate the distance between a query edge image and the template edge image, which are generated from vehicle's possible trajectories and road constraints, respectively. The query image with the smallest distance (or the maximum similarity) will be selected such that the road constraints are considered.

2.4 Place Recognition

Vision-based place recognition problems can be considered as that, given a query image captured at a particular place, return images that depict the same place from the geo-tagged database. In order to perform place recognition, it is necessary to describe the acquired images with a robust, efficient and discriminative descriptor. Thus, the performance of place recognition relies heavily on the descriptor used for

describing different scenes. In this section, we classify place recognition approaches according to the description method employed as: approaches based on hand-crafted descriptors and approaches based on learned descriptors. A graphical description of this classification is shown in Fig. 2.5.

Definition of hand-crafted and learned features:

1. Hand-crafted features: human designed features such as SIFT [78] and SURF [79]. They capture a certain visual property of an image, either globally for the entire image or locally for a given group of pixels.

2. Learned features: deep learning features such as CNN-based features. They learn statistical structure or correlation of an image using deep architectures.

Figure 2.5: Taxonomy for classifying vision-based place recognition approaches according to their image representation method.

2.4.1 Methods Based on Hand-crafted Features

A lot of hand-crafted features have been used to represent an image in the past decades. Most current visual place recognition approaches use point features extracted from corners, blobs and patches to represent a scene and bag-of-word (BOW) based approach, which is first developed for object and image retrieval [80], gains its popularity for its efficiency and robustness. There are usually three steps to represent an image with bag of visual words. In the pre-processing stage, local invariant descriptors are extracted from each image in the database and quantized

into a pre-computed vocabulary of visual words. Using the trained vocabulary, each database image is then represented by a sparse (weighted) frequency histogram of visual words, which can be stored in an efficient inverted file indexing structure. At the query stage, the visual words are firstly extracted from the query image. A short-list of top ranked candidate image in the database is obtained based on the similarity (e.g. cosine similarity, Euclidean and Manhattan distance) between the query and database image BOW vectors. This representation has the good property of easy to work with and a certain robustness to appearance variations caused by illumination changes, lateral shift and dynamic objects.

Based on BOW technique, a localization system called Fast Appearance-Based Mapping (FAB-MAP) is introduced in [81] and SIFT feature is used. To handle problematic situations such as perceptual aliasing, a probabilistic model which can enhance the temporal and spatial consistency of query images is employed. The relevant image with respect to a query scene can be retrieved quickly according to the scores computed from an inverted index scheme. The loop-closure detection results in 70 km and 1000 km trajectories show its effectiveness and robustness. This working principle has become very popular ever since it was introduced.

Similar to visual odometry, feature extraction is also the most time consuming step for a place recognition system. Specific to SIFT feature based FAB-MAP system, computation cost on feature extraction step is around ten times more expensive than the rest of the pipeline. Thus, more computation efficient descriptors are desired for real time loop closure detection. In [82], [83], place recognition framework based on a hierarchical visual bag-of-word model is presented and the combination of very fast image detector–FAST [84] and binary descriptor–BRIEF [85] is used. Since binary descriptors are usually not as robust as non-binary descriptors (e.g. SIFT and SURF) with respect to scale and rotation changes, the performance of binary descriptors on place recognition is questioned. Nevertheless, according to the experiments conducted on real dataset, this feature combination is robust enough for loop closure detection; even for complex situations where mobile vehicle makes

in-plane motion. The second novelty of this framework is the usage of a hierarchical bag of words model together with a direct and standard inverse index scheme, which makes the proposed faster than current approaches. The hierarchical bag of words model exploits hierarchically clustered visual words to pre-define a vocabulary tree [86] which speeds up the searching process than non-hierarchical ones. The standard inverse index is implemented for fast retrieval of images having common visual words with a given one; while the direct index is employed to efficiently obtain point correspondences between images. Due to its real-time performance, place recognizer built on this framework has been used as the loop closing thread in the state-of-art monocular SLAM-ORB-SLAM system [32].

Apart from point features, other popular types of features such as line features have been used for place recognition. Compared with point features, line features carry more structural information since each of them is spanned over a 2D space instead of a single point. Moreover, they are more robust to appearance variations caused by illumination changes, viewing direction changes and occlusions. In [87], the authors propose a place recognition approach using line features and demonstrated better performance than using point features in well structured environments.

On the other end of the spectrum, successful results have also been obtained by using some global feature descriptors which can model the spatial structure and shape of a scene. Compared to point and line features discussed earlier, global descriptors carry the most structure information even though they usually lack the ability to cope with rotation and scale changes. One example is the work of SeqSLAM [88]. Given a query image, it is firstly down-sampled and a single vector describing this scene is constructed. Instead of only using this image, sequences around this image are considered to find the best matching location. Experiments on large scale dataset show its effectiveness even under extreme environment changes. A holistic descriptor vector-Illumination Robust Descriptor (DIRD), which is well-suited for place recognition is proposed in [89]. The authors firstly introduce a fitness function to evaluate a given descriptor in the case of place recognition using holistic features

under varying illumination conditions. And then the illumination robust descriptor is trained and evaluated on an independent test set.

2.4.2 Methods Based on Deep Learned Features

In recent years, Convolutional Neural Networks (CNNs) have been explored and great successes have been achieved. Due to the high level performance, CNN-based methods have been utilized extensively in almost every branch of computer vision. The most active application for CNN must be object detection. Different architectures are used in [90], [91] to get state of the art object detection performance. In [92], [93], CNN models are trained and evaluated on image classification and remarkable results are achieved. The authors of [94], [95] use deep learning method for image retrieval and experiment results show that CNN features outperform classical hand-crafted features. High level performance on semantic segmentation, scene understanding and pose estimation are also achieved by using deep learning method [96], [97].

Since place recognition is essentially a task of image retrieval. It's reasonable to expect CNN-based approach can be applied in place recognition or loop closure detection. As we know, a very large amount of training data is required to train a CNN model (usually with millions of parameters). Hence, how to get a "generic" training data is an issue that urgently needs to be addressed.

Current CNN architecture consists of many layers, including convolution layers, sub-sampling (pooling) layers and fully-connected layers. Given an input image, nonlinear transformations are performed through these layers. And from each layer, one whole image descriptor can be extracted. It is worthy to note that descriptors from different layers have different properties. The high layers tend to encode more semantically meaningful features such as buildings in typical neighbourhood scenes, while lower layers tend to capture low-level image features such as window

corners or roof edges. At the same time, the dimensions of the descriptors vary with the layer depth. In image classification application, the descriptors from final layer are commonly used to encode the contents of the image in a probability distribution form. However, in place recognition application, high degree of robustness (to illumination, view point changes and dynamic objects), as well as high degree of distinctiveness (to similar scenes) is required. Topmost layer is not necessarily the best option. Hence, find which layer is the most appropriate for visual place recognition is another issue to be resolved.

Of course , there are other issues, like can the model trained for image classification be used for place recognition and is a model trained for urban area place recognition generic enough for sub-urban area place recognition?

In recent five years, place recognition frameworks based on CNN models have been explored and most of them are aimed at solving the above discussed issues. In [98], a pre-trained CNN network called Overfeat, which was originally proposed for the ImageNet Large Scale Visual Recognition Challenge 2013 [92] has been used to learn CNN features. In order to enforce the spatial and temporal consistency of the test images, a spatial and sequential filter is utilized. Loop closures in challenge datasets are detected using these powerful features. Comprehensive comparison experiments between CNN learned features from different layers and conventional features are conducted. From their experiment results, the authors conclude that a supervised deep CNN model trained for other task like image classification, can be used for place recognition. At the meanwhile, different layers appear to be suitable for different scenarios of place recognition. For relatively static, similar viewpoint datasets, the middle layers are optimal. But when viewpoint variance becomes significant, higher layers perform better. Similar explorations are also conducted in [99]. One difference should be noticed that, a CNN model trained from a scene-centric dataset called Places [100] with over 2.5 million images of 205 scene categories has been used to learn CNN features. Relative detailed comparisons between CNN features and hand-crafted features are made. The authors concluded that CNN-based image

descriptors perform similarly to hand-crafted descriptors in environments without illumination change, but outperform hand-crafted descriptors significantly when the environment undergoes obvious illumination changes.

The above studies prove that CNN-based image descriptors can be used for place recognition purpose and perform even better in challenging environments. However, there are still some issues need to be addressed. The CNN models used to learn CNN descriptors in the above cases are not trained specifically for place recognition task. So one would ask whether the performance can be further improved by using CNN models trained on place recognition dataset. Another problem is automatic layer selection. While similar conclusions are obtained that different layers perform differently for place representation, it is still unclear how to automatically select the best layer for a specific place recognition task. So challenges and opportunities still exist for further research in this area.

2.5 Conclusions

This chapter has presented the relevant background of vision based localization and gives an extensive survey. The principles of vision-based localization approaches which include visual odometry, visual SLAM and place recognition are introduced. Besides, localization methods using digital maps are depicted. The main approaches published in the last ten yeas with regard to our common thread are reviewed. As typical metric localization approaches, visual odometry and visual SLAM are playing increasing important roles in mobile vehicle navigation. The common drawback for them is the positioning drift after long range navigation. And decreasing the accumulated error is one of the directions of current efforts. Distinguished from visual odometry and visual SLAM, place recognition belongs to topological category due to its discrete nature. Although, tremendous success have been made, reliable place recognition is still a tough problem, especially in large-scale outdoor environments.

Appearance variation, repetitive structure, perceptual aliasing, scene dynamics and etc. are the open challenges. The common thread of the following chapters is to develop robust and efficient vision-based localization techniques that can cope these challenges for unmanned vehicles.

Chapter 3

Road Constrained Monocular Visual Localization Using Gaussian-Gaussian Cloud Model

3.1 Introduction

In recent years, vision-based navigation has drawn significant attention due to their low cost and possibility to provide a full 6 DOF motion estimation. Visual odometry, as the most representative method, is the process of estimating the ego-motion of the camera. It has been used for mobile robot localization thanks to their constantly improving performance. However, with this kind of approach, the small errors of motion estimation accumulate as the vehicle moves. This issue has prevented visual odometry from being implemented in long range navigation. Compared with stereo visual odometry (SVO), monocular visual odometry (MVO) is more popular on small scale robots due to its lower cost and higher performance with short baseline. Nevertheless, it is challenging to estimate the vehicle's true position using MVO due to the scale ambiguity issue.

To tackle the above-mentioned drift issue of VO, current efforts are mainly focusing on incorporating VO into a Simultaneous Localization And Mapping (SLAM) system [32]. A loop closing thread is usually implemented to detect loops and address accumulated drifts in these methods. Translational and rotational drifts as well as scale drifts in MVO can be reduced by pose graph optimization when loop closing happens. While good performance can be achieved, the requirement of driving in loops limits the usage of this kind of method in real driving conditions. To obtain the metric scale of MVO, other sensors like IMU [17] or objects with known size [101] are usually utilized to provide extra information about the absolute scale. Another popular method is assuming that the camera is moving at a known and fixed height over the ground [20]. However, the result of this kind of method is largely based on the accuracy of the ground plane detection.

Both drift and scale ambiguity are part of measurement uncertainties. Thus, in order to reduce drift and scale ambiguity in MVO, it is necessary to discuss measurement uncertainties. Generally speaking, randomness and fuzziness are the two most widely used terms to describe measurement uncertainties. Probability theory and fuzzy mathematics have been respectively utilized to model them. However, in many kinds of measurements, these two uncertainties co-exist and it is hard to distinguish them. A group of researchers proposed a so-called "cloud model" in [54]. Instead of considering randomness and fuzziness separately, this model provides a unified representation. Inspired by this, a Gaussian-Gaussian Cloud (GGC) model is proposed in this chapter. The drift and scale ambiguity of MVO are both considered as measurement uncertainties and are incorporated into the presented GGC model.

Benefited from the development of digital maps, map-assisted positioning has provided new ideas for mobile vehicle localization [72], [102]. Particularly, the geometric shape of road network is fixed and often unique in a certain region. By assuming a vehicle moves on a drivable road, its historical trajectory would correspond to a route on the map. Then, finding the route whose shape best aligns the historical

trajectory on the map will give constraints to limit the vehicle's possible position; thus localization accuracy could be increased.

In this chapter, we present a framework which aims at globally localizing a mobile vehicle equipped with one single camera and a freely available OpenStreetMap (OSM). "Cloud drops" are generated according to MVO measurement based on GGC model. Road constraints are incorporated into the localization framework to filter out cloud drops which are inconsistent with the digital map. With the integration of road constraints, the autonomous vehicle can be localized in a more efficient and accurate way. The main contributions of this chapter are as follows:

1. Cloud model is used to describe the uncertainty of raw measurement of visual odometry. Specifically, Gaussian-Gaussian Cloud is firstly proposed to represent scale ambiguity and measurement randomness in monocular visual odometry.

2. Road constraints from the geometric shapes of road network are utilized to reduce measurement uncertainties.

3. A parameter estimation scheme is presented to further narrow down scale ambiguity while resampling cloud drops.

4. Comprehensive evaluations with practical data have been conducted to verify the effectiveness of the proposed localization framework.

The remaining part of the chapter proceeds as below. The problem we are trying to resolve is formulated in Section 3.2. The concept of Gaussian-Gaussian Cloud and their characteristics are depicted in Section 3.3. Section 3.4 describes the localization framework, where the details of all the steps are emphasized. In Section 3.5, experiments on KITTI benchmark and our own dataset are implemented and the results are discussed. Section 3.6 concludes the chapter.

3.2 Problem Formulation

The autonomous vehicle equipped with visual odometry takes images with a rigidly-attached camera system at each discrete time instant k [16]. By feature detection, feature matching, motion estimation and local optimization, the visual odometry outputs a rigid body raw transformation matrix between time instant k and $k-1$ for every k:

$$\mathbf{T}_{k,k-1}^{\text{raw}} = \begin{bmatrix} \mathbf{R}_{k,k-1}^{\text{raw}} & \mathbf{t}_{k,k-1}^{\text{raw}} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{3.1}$$

where $\mathbf{R}_{k,k-1}^{\text{raw}} \in SO(3)$ and $\mathbf{t}_{k,k-1}^{\text{raw}} \in \mathbb{R}^3$ denote the rotation and translation parts of $\mathbf{T}_{k,k-1}^{\text{raw}}$, respectively.

Given the raw transformation matrix, the raw camera pose $\mathbf{C}_k^{\text{raw}} = \left[\mathbf{R}_k^{\text{raw}} | \mathbf{t}_k^{\text{raw}}\right]$ can be computed by the raw measurement equation

$$\mathbf{C}_k^{\text{raw}} = \mathbf{C}_{k-1}^{\text{raw}} \mathbf{T}_{k,k-1}^{\text{raw}}. \tag{3.2}$$

For SVO, the above estimated translation $\mathbf{t}_{k,k-1}^{\text{raw}}$ is at the true scale. However, for the monocular case, due to the purely projective nature of a single camera, the motion can only be estimated up to an unknown scale. Moreover, scale drift exists in monocular odometry, which means the absolute scale varies as the camera moves. Thus, the true camera pose $\mathbf{C}_k = \left[\mathbf{R}_k | \mathbf{t}_k\right]$ at every time instant can be represented by a more generalized scaled measurement equation

$$\mathbf{C}_k = \mathbf{C}_{k-1}(\mathbf{s}_k \otimes \mathbf{T}_{k,k-1}^{\text{raw}}), \tag{3.3}$$

where the scaled transformation matrix $\mathbf{s}_k \otimes \mathbf{T}_{k,k-1}^{\text{raw}}$ is defined as

$$\mathbf{s}_k \otimes \mathbf{T}_{k,k-1}^{\text{raw}} = \begin{bmatrix} \mathbf{R}_{k,k-1}^{\text{raw}} & \mathbf{s}_k \circ \mathbf{t}_{k,k-1}^{\text{raw}} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{3.4}$$

in which $\circ$ denotes the Hadamard product (or the Schur product); $\mathbf{s}_k$ is the scaling vector at time instant k with compatible dimensions. The above scaled measurement equation can be applied to both MVO and SVO. For ideal SVO, Eqn. (3.3) can be specified by setting $\mathbf{s}_k$ as $\mathbf{1} = [1, 1, 1]^T$.

By defining $\mathbf{t}_{k,k-1} = \mathbf{s}_k \circ \mathbf{t}^{\text{raw}}_{k,k-1}$, Eqn. (3.4) can be written as

$$\mathbf{s}_k \otimes \mathbf{T}^{\text{raw}}_{k,k-1} = \begin{bmatrix} \mathbf{R}^{\text{raw}}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{3.5}$$

which can be incorporated into Eqn. (3.2). As in most localization problems, the translation vector $\mathbf{t}_k$ is more concerning compared to the rotation matrix $\mathbf{R}_k$, this chapter focuses on measurement uncertainties representation with regard to $\mathbf{t}_{k,k-1}$. Then, the localization problem can be formulated as follows:

Problem. *Given the initial camera pose $\mathbf{C}_0$ and the raw transformation matrix $\mathbf{T}^{raw}_{k,k-1}$ obtained from VO, estimate the scaled camera pose $\mathbf{C}_k$ at each time k with the least possible measurement error e_k, which is defined as $e_k = ||\mathbf{t}_{k,k-1} - \bar{\mathbf{t}}_{k,k-1}||$, where $\bar{\mathbf{t}}_{k,k-1}$ denotes the true translation vector at time instant k.*

The error of $\mathbf{t}_{k,k-1}$ comes from mainly two sources: error of $\mathbf{t}^{\text{raw}}_{k,k-1}$ and error of $\mathbf{s}_k$. The error of $\mathbf{t}^{\text{raw}}_{k,k-1}$ comes from measurement randomness, which can be properly modelled by conventional probability theory (e.g. Gaussian probability assumption). However, it is difficult to describe the error of $\mathbf{s}_k$ in a similar way since there is no prior knowledge of $\mathbf{s}_k$ and MVO does not provide any measurement of it. Thus the scale ambiguity exists.

Cloud model has been proposed to represent uncertainties by combining randomness and fuzziness together. In this chapter, the concept of cloud is utilized for VO measurement representation and the details are explained in the next sections. For the sake of easy expression, several reasonable assumptions are made as follows:

Assumption 1. *In MVO measurement Eqn. (3.3), for every time instant k, ele-*

ments in $\mathbf{s}_k$ *and* $\mathbf{t}_{k,k-1}^{raw}$ *are mutually independent.*

Assumption 2. *The raw translation vector* $\mathbf{t}_{k,k-1}^{raw}$ *and the scaling vector* $\mathbf{s}_k$ *both obey Gaussian distribution with corresponding expectations and variances.*

3.3 Cloud Model for Measurement Uncertainties

As discussed in previous section, the drift and scale factor should be modelled properly in a monocular odometry localization system. Analysing the distribution of the scaled translation, we found that it's no longer a Gaussian and can't be modelled with a commonly used parametric distribution. Hence, we turn to other theories. A decade ago, Li *et al* [54],[55] proposed a new concept called cloud model for concept conversion between qualitative and quantitative ones. In this section, the concept of cloud is firstly explained. Then, a new type of cloud (Gaussian-Gaussian Cloud, GGC) is defined and its statistical characteristics are discussed. The VO measurement representation based on GGC is described later. Lastly, we present a parameter estimation scheme for GGC.

3.3.1 Cloud and Gaussian-Gaussian Cloud

Definition 3.1 (Cloud [54])**.** *Let* U *be a universal set described by a precise number, and* C *be the qualitative concept related to* U*. If there is a number* $x \in U$*, which randomly realizes the concept* C*, and the certainty degree of* x *for* C*, i.e.,* $\mu(x) \in [0, 1]$*, is a random value with stable tendency:*

$$\mu : U \to [0, 1] \quad \forall x \in U \quad x \to \mu(x). \tag{3.6}$$

Then the distribution of x *on* U *is defined as a* ***cloud*** $C(x)$*, and every* x *is defined as a* ***cloud drop****. The certainty degree* μ *also refers as a membership grade.*

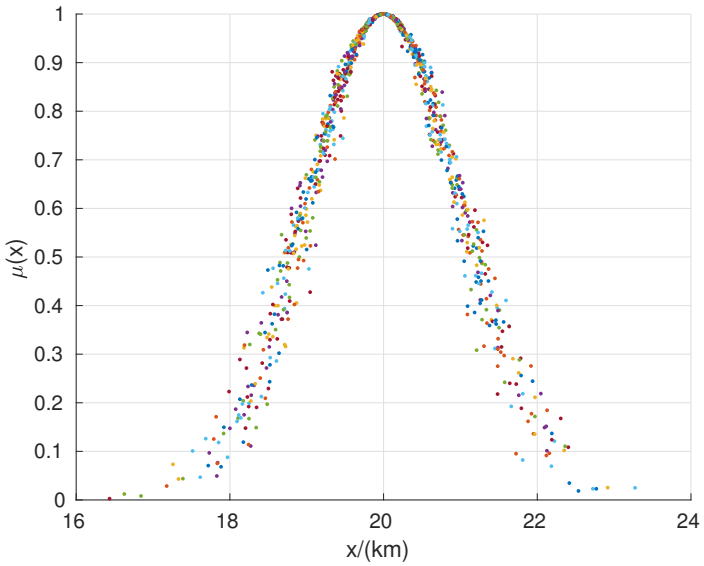


Figure 3.1: This figure illustrates a Gaussian cloud generated from 1000 cloud drops for the concept "distance around 20 km". x is a distance around 20 km and each x is a cloud drop; $Ex = 20$ km, $En = 1$ km, $He = 0.1$ km; μ is the certainty degree of x for the concept "distance around 20 km".

By incorporating randomness and fuzziness into unified consideration, the cloud model is able to represent concepts with uncertainty, where each cloud drop contribute to the concept with a particular "degree of contribution". Since Gaussian is one of the most important distribution, the authors of [54] further define a Gaussian cloud model based on the above definition. In the Gaussian cloud model, three numerical characteristics (expected value Ex, entropy En and hyper entropy He) of the quantitative number x are defined to describe the given qualitative concept C. Mathematically, if $x \sim N(Ex, En'^2)$, $En' \sim N(En, He^2)$ and certainty degree of x for C satisfies $\mu(x) = \exp(-\frac{(x-Ex)^2}{2*En'^2})$, then the distribution of x is called Gaussian cloud. Fig. 3.1 demonstrates one Gaussian cloud for the concept "distance around 20 km". As can be seen, a cloud drop is nothing but a random realization of the qualitative concept. The certainty degree for each cloud drop is a random value with stable tendency-when the distance x tends to 20 km, the corresponding certainty degree $\mu(x)$ tends to 1 and when x is far away from the concept, it's certainty degree degrades.

Although the Gaussian cloud model is the most-studied cloud model as its universality, different types of clouds can be created according to different probability distributions, such as uniform cloud, power law cloud, trapezium cloud etc. The appropriate choice of cloud model is based on accurate comprehension of the physical process that describes the designated problem. In this work, in order to properly model uncertainties in VO measurement we define Gaussian-Gaussian Cloud as follows:

Definition 3.2 (Gaussian-Gaussian Cloud, GGC)**.** *Let U be the universe of discourse and GGC be a qualitative concept in U. If $\mathbf{Y} = \mathbf{S} \circ \mathbf{X} \in U$ is a random instantiation of concept GGC, where $\mathbf{S}$ and $\mathbf{X}$ are random vectors that obey independent Gaussian distribution $N(E\mathbf{S}, \Sigma_S)$ and $N(E\mathbf{X}, \Sigma_X)$, respectively, and the certainty degree of $\mathbf{Y}$ belonging to concept GGC satisfies (3.7), where diag$(\mathbf{S}^T)$ de-*

$$\mu(\mathbf{Y}) = \exp\left\{-\frac{1}{2}\left[\text{diag}(\mathbf{S}^T)^{-1}\mathbf{Y} - E\left(\mathbf{X}\right)\right]^T \Sigma_X^{-1}\left[\text{diag}(\mathbf{S}^T)^{-1}\mathbf{Y} - E\left(\mathbf{X}\right)\right]\right\} \tag{3.7}$$

notes the diagonal matrix with corresponding diagonal entries from $\mathbf{S}^T$, then the distribution of $\mathbf{Y}$ in the universe U is a multivariate Gaussian-Gaussian cloud, which can be denoted by $\mathbf{Y} \sim GGC(E\mathbf{S}, \Sigma_S, E\mathbf{X}, \Sigma_X)$.

Remark 1. *It is evident that the certainty degree of $\mathbf{Y}$ is a random value for any fixed $\mathbf{Y}$, thus the distribution of $\mathbf{Y}$ accords with the cloud model.*

Remark 2. *Definition 3.2 generalizes the definition in [54] by considering the multi-dimension of random vectors $\mathbf{S}$ and $\mathbf{X}$. Definition 3.2 also generalizes the conventional Gaussian distribution: If $\Sigma_S = \mathbf{0}$, then $\mathbf{S}$ will be a fixed vector, which results in a multivariate bell-shaped distribution.*

Given the parameters of a GGC, cloud drops and their corresponding certainty degrees can be generated based on Algorithm 1.

Algorithm 1 GGC Generator

Input: GGC parameters $E\mathbf{S}, \Sigma_S, E\mathbf{X}, \Sigma_X$, number of cloud drops N.
Output: Cloud drops and corresponding certainty degrees $(\mathbf{y}_i, \mu_i), i \in \{1, ..., N\}$

for $i \leftarrow 1$ to N **do**
  $\mathbf{s}_i \leftarrow \text{NORM}(E\mathbf{S}, \Sigma_S)$ /**generating a Gaussian-distributed random vector* $\mathbf{s}_i$*/
  $\mathbf{x}_i \leftarrow \text{NORM}(E\mathbf{X}, \Sigma_X)$ /**generating a Gaussian-distributed random vector* $\mathbf{x}_i$*/
  $\mathbf{y}_i = \mathbf{s}_i \circ \mathbf{x}_i$
  $\mu_i \leftarrow \exp\left[-\frac{1}{2}(\mathbf{x}_i - E\mathbf{X})^T \Sigma_X^{-1} (\mathbf{x}_i - E\mathbf{X})\right]$ /**calculating the certainty degree of* $\mathbf{y}_i$*/
end for

3.3.2 Visual Odometry Measurement Representation Based on Gaussian-Gaussian Cloud

As discussed in previous sections, the concept of cloud can be used to represent measurement with uncertainties. In this section, we represent VO measurement with aforementioned GGC. Based on Assumption 2, it is natural to define $\mathbf{s}_k$ and $\mathbf{t}_{k,k-1}^{\text{raw}}$ as random vectors $\mathbf{S}$ and $\mathbf{X}$ which satisfy $\mathbf{S} \sim N(E\mathbf{S}, \Sigma_S)$ and $\mathbf{X} \sim N(E\mathbf{X}, \Sigma_X)$ respectively, where $N(E, \Sigma)$ denotes Gaussian distribution with expectation E and covariance matrix Σ. Then, according to Definition 3.2, the scaled measurement follows $\mathbf{t}_{k,k-1} \sim GGC(E\mathbf{S}, \Sigma_S, E\mathbf{X}, \Sigma_X)$.

The *GGC* generator described in Algorithm 1 can be implemented to generate a cloud with certain parameters and a given cloud drop number. For each cloud drop $\mathbf{y}_i$, there is a unique certainty degree $\mu(\mathbf{y}_i)$ which indicates the degree that the cloud drop is able to represent raw measurement. Particularly, $\mu(\mathbf{y}_i) = 1$ if and only if $\mathbf{x}_i = E\mathbf{X}$; $\mu(\mathbf{y}_i) \to 0$ if and only if $|\mathbf{x}_i - E\mathbf{X}| \to \infty$. In other words, the cloud drop $\mathbf{y}_i$ is more representative for raw measurement if $\mu(\mathbf{y}_i)$ is larger.

As the sources of positioning error, both scale ambiguity and measurement randomness should be considered when representing visual odometry measurement. By using the concept of GGC, the two major uncertainties are integrated and represented in a single model, such that drops can be generated as realistic as possible in visual odometry. With this model, measurement from both stereo visual odometry and monocular visual odometry can be properly modelled by combining the drift

and scale ambiguity in an intuitive way. Specifically, scale ambiguity and measurement randomness are represented with random vectors $\mathbf{S}$ and $\mathbf{X}$, respectively. The influence of parameters on GGC is summarized as follows: $E\mathbf{S}$ gives an estimation for the average scale; Σ_S measures the degree of scale ambiguity; $E\mathbf{X}$ reflects raw measurement result; Σ_X represents the degree of raw measurement randomness. Fig. 3.2 demonstrates the influence of parameters on a one-dimensional GGC in a visualized way.

3.3.3 Statistical Analysis for Gaussian-Gaussian Cloud

We now move on and discuss several statistical characteristics of GGC including probability density function, moments and certainty degree distribution. In the following content, we assume $\mathbf{Y} \sim GGC(E\mathbf{S}, \Sigma_S, E\mathbf{X}, \Sigma_X)$ and we use Y_j to represent the j-th random variable in random vector $\mathbf{Y}$ for the sake of analysis. These properties will be employed for parameter estimation.

Property 1 (PDF of GGC)**.** *The Probability Density Function (PDF) of Y_j [103] can be represented as (3.8)*

where $\sigma_{S_j}^2$ and $\sigma_{X_j}^2$ denote the j-th diagonal element in covariance matrices Σ_S and Σ_X, respectively.

Property 2 (Expectation)**.** *The expectation or the 1st moment of Y_j and $\mathbf{Y}$ can be obtained from*

$$E(Y_j) = E(S_j)E(X_j), \tag{3.10}$$

$$E\mathbf{Y} = E\mathbf{S} \circ E\mathbf{X}. \tag{3.11}$$

Property 3 (Variance)**.** *The variance or the 2nd central moment of Y_j can be*

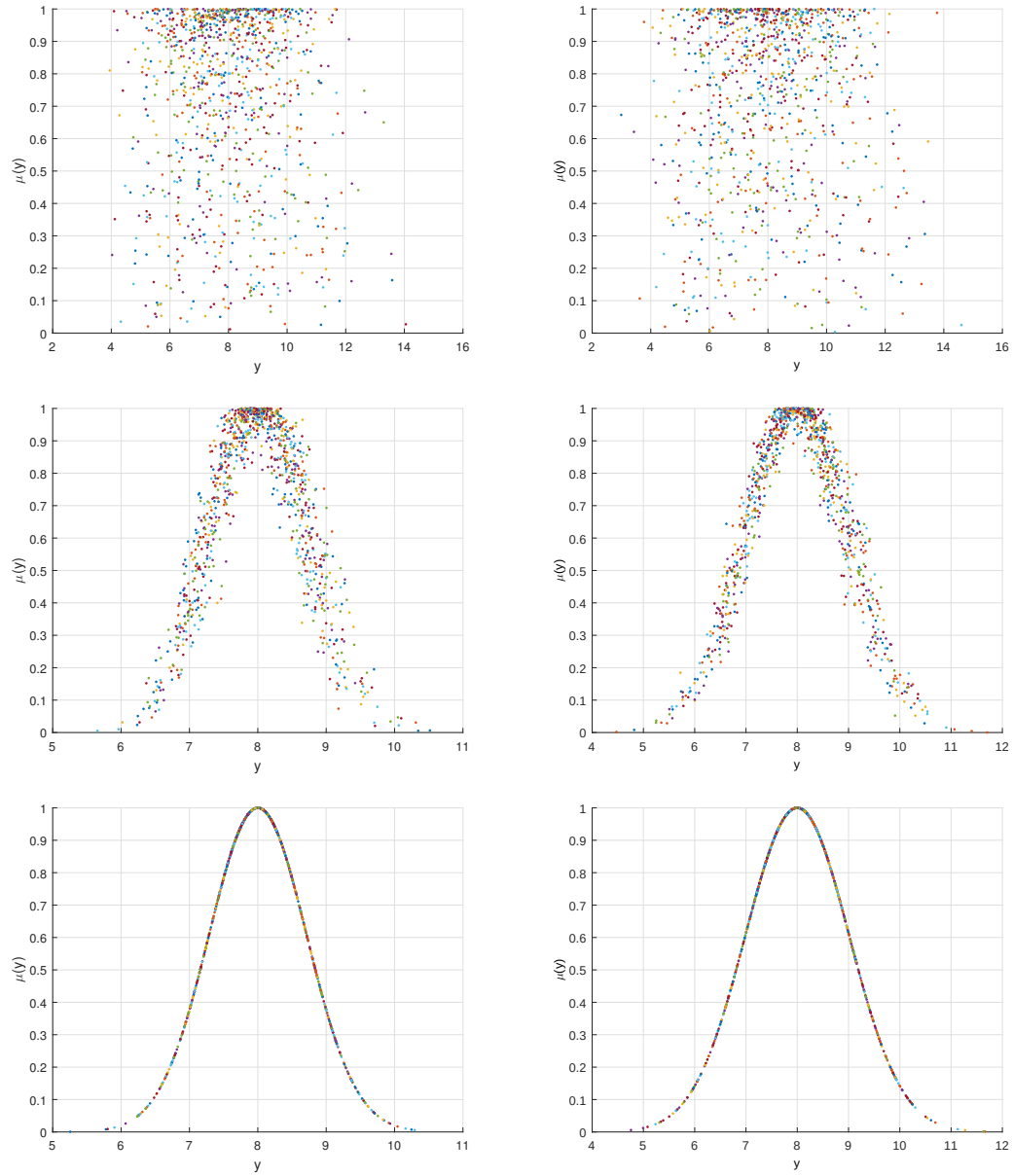


Figure 3.2: One-dimensional GGC examples $C_{11}(1.6, 0.1, 5, 0.2)$, $C_{12}(1.6, 0.1, 5, 0.4)$, $C_{21}(1.6, 0.001, 5, 0.2)$, $C_{22}(1.6, 0.001, 5, 0.4)$, $C_{31}(1.6, 0, 5, 0.2)$, and $C_{32}(1.6, 0, 5, 0.4)$, each with 1000 cloud drops. Up: cloud C_{11} and C_{12} show strong scale ambiguity, as cloud drops with similar y lead to distinct μ. Middle: cloud C_{21} and C_{22} tends to converge to Gaussian distribution with smaller Σ_S. Down: cloud C_{31} and C_{32} fit perfectly to Gaussian distribution when $\Sigma_S = 0$.

obtained from

$$D(Y_j) = D(X_j)E(S_j)^2 + D(S_j)E(X_j)^2 + D(S_j)D(X_j) \tag{3.12}$$

$$f_{Y_j}(y) = \int_{-\infty}^{+\infty} f_{S_j}(s) f_{X_j}(\frac{y}{s}) \frac{1}{|s|} \mathrm{d}s \tag{3.8}$$

$$= \frac{1}{2\sigma_{S_j}\sigma_{X_j}\pi} \int_{-\infty}^{+\infty} \exp\left\{\frac{[s - E(S_j)]^2}{-2\sigma_{S_j}^2}\right\} \exp\left\{\frac{[y/s - E(X_j)]^2}{-2\sigma_{X_j}^2}\right\} \frac{1}{|s|} \mathrm{d}s, \tag{3.9}$$

$$= \sigma_{X_j}^2 E(S_j)^2 + \sigma_{S_j}^2 E(X_j)^2 + \sigma_{S_j}^2 \sigma_{X_j}^2. \tag{3.13}$$

Property 4 (PDF of Cloud Drop Certainty Degree Distribution)**.** *The probability distribution of* $\mu(\mathbf{Y})$ *in GGC can be represented as*

$$f_M(\mu) = \begin{cases} \frac{(1/2)^{\frac{P}{2}-1}}{\Gamma(P/2)} (-2\ln\mu)^{\frac{P}{2}-1} & \text{if } \mu \in (0,1) \\ 0 & \text{otherwise} \end{cases}, \tag{3.14}$$

where Γ *denotes the gamma function, and* $P \in \mathbb{Z}^+$ *is the dimension of* $\mathbf{Y}$*.*

Proof According to Eqn. (3.7), let D be a random variable that obeys cloud drop certainty degree distribution, then each cloud drop certainty degree μ_i can be regarded as a sample generated from the certainty degree distribution. The cumulative distribution function of D can be represented as

$$F_D(\mu) = \mathbb{P}\{D \le \mu\} \tag{3.15}$$

$$= \mathbb{P}\left\{\exp\left[-\frac{1}{2}(\mathbf{X} - E\mathbf{X})^T \Sigma_X^{-1} (\mathbf{X} - E\mathbf{X})\right] \le \mu\right\} \tag{3.16}$$

$$= \mathbb{P}\left\{\sum_{j=1}^{P} \left[\frac{X_j - E(X_j)}{\sigma_{X_j}}\right]^2 \ge -2\ln\mu\right\}. \tag{3.17}$$

As $X_j \sim N(E(X_j), \sigma_{X_j}^2)$, we have $(X_j - E(X_j))/\sigma_{X_j} \sim N(0,1)$. Based on the definition of χ^2-distribution, it is noticed that

$$\sum_{j=1}^{P} \left[\frac{X_j - E(X_j)}{\sigma_j}\right]^2 \sim \chi^2(P), \tag{3.18}$$

where P denotes the dimension of $\mathbf{Y}$. Then (3.17) becomes

$$F_D(\mu) = \int_{-2\ln\mu}^{+\infty} f_{\chi^2,P}(t)\mathrm{d}t \tag{3.19}$$

$$= \int_{-2\ln\mu}^{+\infty} \frac{(1/2)^{\frac{P}{2}}}{\Gamma(P/2)} t^{\frac{P}{2}-1} \exp\left(-\frac{t}{2}\right) \mathrm{d}t. \tag{3.20}$$

The PDF of D can be obtained from

$$f_{M_j}(\mu) = F'_D(\mu) \tag{3.21}$$

$$= -\frac{(1/2)^{\frac{P}{2}}}{\Gamma(P/2)}(-2\ln\mu)^{\frac{P}{2}-1} \exp(\ln\mu)(-2\ln\mu)' \tag{3.22}$$

$$= \frac{(1/2)^{\frac{P}{2}-1}}{\Gamma(P/2)}(-2\ln\mu)^{\frac{P}{2}-1}. \tag{3.23}$$

Remark 3. *The distribution of μ is independent of GGC parameters. This implies that even though there are differences in understanding the same concept for each individual, the overall cognitive rule is consistent, which is one important character of cloud model. Particularly, for $GGC(E\mathbf{S}, \Sigma_S, E\mathbf{X}, \Sigma_X)$, we may conclude that the cloud drop certainty degree μ in GGC with a certain dimension always obeys same distribution regardless of $E\mathbf{S}$, Σ_S, $E\mathbf{X}$ and Σ_X.*

3.3.4 Parameter Estimation for Gaussian-Gaussian Cloud

The term parameter estimation refers to the process of using sample data to estimate the parameters of the selected distribution. The parameter estimation problem for GGC is defined as follows: Given a cloud drop set $\mathcal{S} = \{\mathbf{y}_i\}$, find $\hat{E}\mathbf{S}, \hat{\Sigma}_S, \hat{E}\mathbf{X}, \hat{\Sigma}_X$ such that the cloud drop set $\hat{\mathcal{S}}$ generated from $GGC(\hat{E}\mathbf{S}, \hat{\Sigma}_S, \hat{E}\mathbf{X}, \hat{\Sigma}_X)$ is *most similar* to $\mathcal{S}$.

The phrase *most similar* can be explained from different perspectives, which lead to different estimation methods. In this particular problem, it is difficult to implement the maximum likelihood estimator since the PDF of GGC is expressed with integrals.

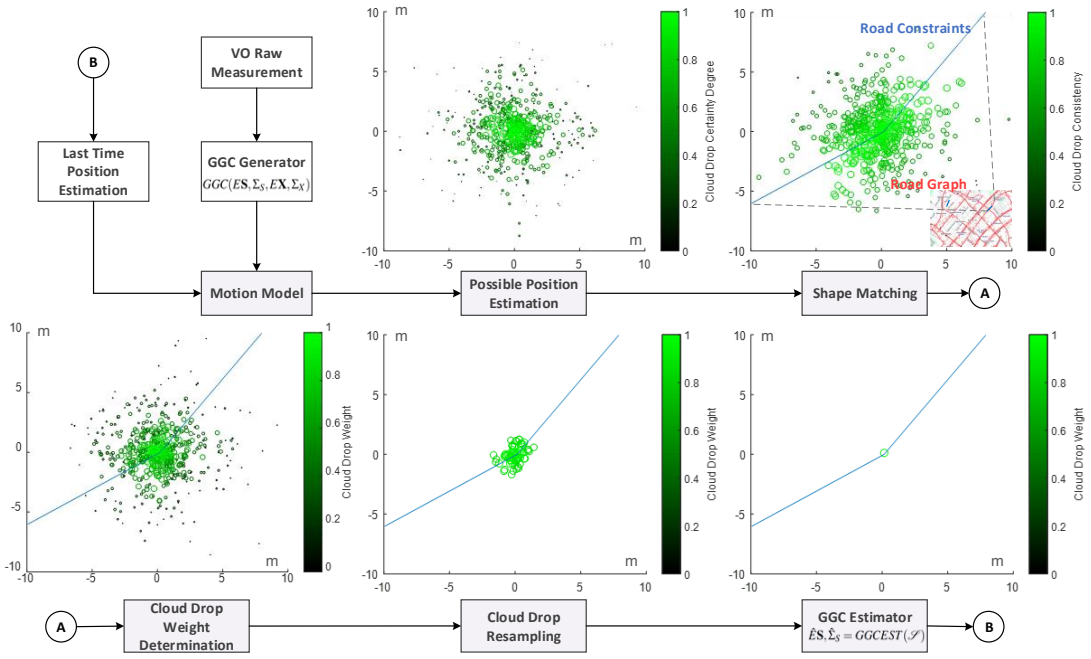


Figure 3.3: Framework of the road constrained localization approach based on GGC.

In statistics, the method of moments is a widely used approach to estimate the parameters of a population. Based on the above statistical analysis, the relationships between the moments and the parameters of GGC are described by Eqn. (3.10-3.12). Given these equations and sample moments of a set of cloud drops, the parameters of interest can be estimated by solving these equations. Theoretically, to estimate the four parameters, the first four moments of the cloud drop set $\mathcal{S}$ are required to establish a higher-order system as follows, from which algebraic or numerical result can be solved:

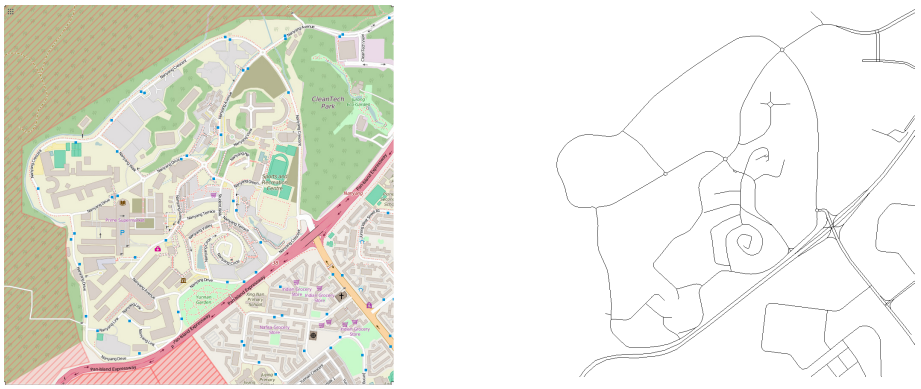
$$\mathbf{M}_{\mathrm{GGC}}(\hat{E}\mathbf{S}, \hat{\Sigma}_S, \hat{E}\mathbf{X}, \hat{\Sigma}_X) = \mathbf{M}_{\mathrm{cldrp}}(\mathcal{S}), \tag{3.24}$$

where $\mathbf{M}_{\mathrm{GGC}}(\hat{E}\mathbf{S}, \hat{\Sigma}_S, \hat{E}\mathbf{X}, \hat{\Sigma}_X)$ and $\mathbf{M}_{\mathrm{cldrp}}(\mathcal{S})$ denote the moments expressed with GGC parameters and sample moments, respectively.

3.4 Monocular Localization with Cloud Model

As described in Section 3.3.2, the odometry measurement can be represented with a GGC model. To complete our localization objective, a road-constrained localization approach is proposed and the framework is demonstrated in Fig. 3.3. As can be seen, similar to a Monte Carlo Localization system, motion update step as well as observation step all show up in this framework. Visual odometry plays a role of motion model while the shape matching process plays a role of measurement model. Each possible state of the vehicle at every time step is represented with a cloud drop in our Gaussian-Gaussian Cloud. After each time cloud drops are updated, shape matching will be implemented to evaluate the similarity between geometric map and the travelled path by examining each possible trajectory consists of historical and current estimations. The localization process can be considered as the converging process of cloud drops towards the actual position. To narrow down the range of the scale and reduce the computational load, a scale ambiguity reduction scheme is conducted at the end of each iteration.

3.4.1 Map Preprocessing



(a) OSM. (b) Road graph.

Figure 3.4: The raw OpenStreetMap and the preprocessed road graph of Nanyang Technological University.

We use the openly licensed OpenStreetMap as our map. The OpenStreetMap is freely available, easily editable and frequently updated. In this map preprocessing step, a coarse positioning result such as a very rough GPS (can be a few hundred meters away from the true position), is desired so that the map of a certain region can be downloaded. If the rough position is not available, any other position indication such as a block name will do as long as the approximate position range of the vehicle can be determined.

In order to use the downloaded map (Fig. 3.4a) in a concise way, several preprocessing operations are deployed. At first, all the semantic information, such as buildings, traffic signs, tracks, etc. in the map are eliminated. Only crossings and drivable roads connecting them are preserved and used to form a road graph, as demonstrated in Fig. 3.4b. After the road graph is extracted, an edge detection step is performed to convert road graph to a template edge map, which will be utilized in shape matching step.

3.4.2 Road Constrained Shape Matching

We assume that the vehicle is always moving on a drivable road. The road shape provides constraints to the vehicle's possible trajectory. And this information is used to correct VO trajectory in this chapter. Shape matching is an algorithm in computer vision area. It is designed to find the best alignment between two edge maps. If we convert road map and historical trajectory of each drop to a template and query edge map respectively, then the shape matching result gives a measure of matching degree between the two edge maps.

More specifically, to use road constraints effectively, we first convert the road graph to a template edge map $M = \{m_k\}$, where m_k is one edge segment. All the drivable roads in map are now represented by edges in M. The historical trajectory of each drop is plotted and also converted to a query edge map $Q = \{q_l\}$, where q_l is one

edge point. Then the shape matching distance between M and Q is given by the average distance, including Euclidean and orientation difference between each point $q_l \in Q$ and its nearest edge in M [104]

$$d_{\mathrm{SM}}(Q, M) = \frac{1}{n} \sum_{q_l \in Q} \min_{m_k \in M} \{|q_l - m_k| + \alpha |O_{q_l} - O_{m_k}|\}, \tag{3.25}$$

where n is the total number of points in Q; O_{q_l} is the tangent direction of point q_l; O_{m_k} is the orientation of edge m_k; α is the weight of orientation distance.

A *consistency* metric $\upsilon(\mathbf{y}_i)$ needs to be defined for each cloud drop to denote what degree the drop accord with road constraints. Moreover, a fittest function modelling the relationship between the above-computed shape matching distance d_{SM} and the *consistency* $\upsilon(\mathbf{y}_i)$ needs to be obtained. In this chapter, we use some sample data to compute the drop's empirical *consistency* distribution with regard to shape matching distance. And some parametric distributions are utilized to fit the empirical one afterwards. Concretely, for a real road on the map, a set of pseudo trajectories is generated to simulate vehicle's possible trajectories. The similarity between each pseudo trajectory and the real trajectory is used to represent the *consistency*. And the shape matching distance is calculated from Eqn. (3.25). The fitting result is demonstrated in Fig. 3.5. Noted that the empirical distribution closely follows a Chi-squared distribution, $\upsilon(\mathbf{y}_i)$ can be expressed as

$$\upsilon(\mathbf{y}_i) = \chi(d_{\mathrm{SM}}, k), \tag{3.26}$$

where $\chi(d_{\mathrm{SM}}, k)$ denotes the estimated Chi-squared distribution with k degrees of freedom; d_{SM} is the shape matching distance, which is determined by the difference between the trajectory to be evaluated and the geometric map: The more geometric similarities are there, the lower d_{SM} is.

Since it is supposed that the *consistency* is negatively associated with the shape matching distance, one would think that the ideal distribution of d_{SM} should follow

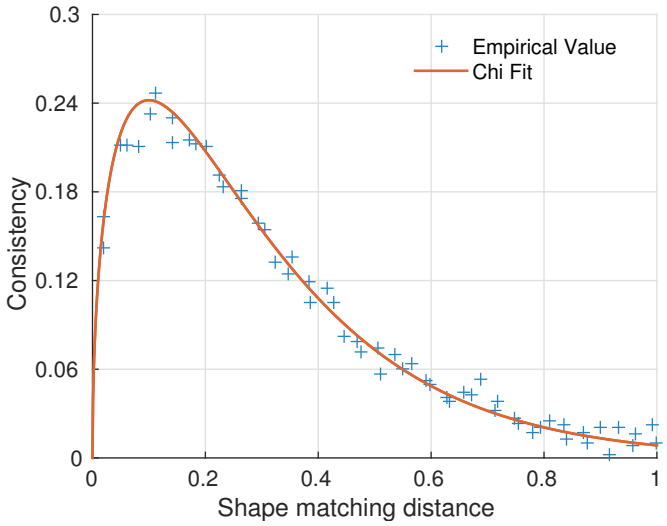
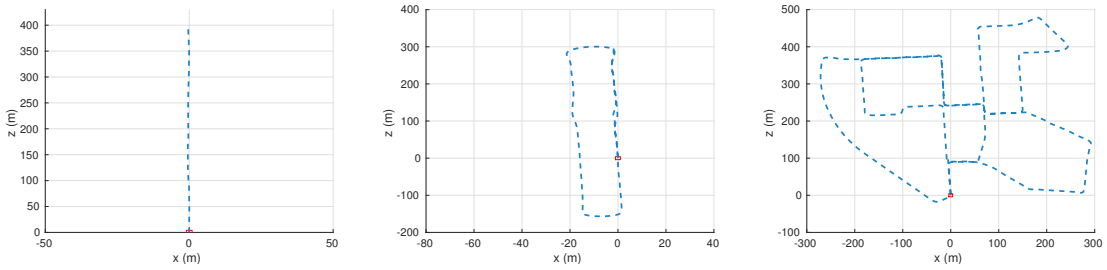


Figure 3.5: Drop consistency distribution with regard to shape matching distance.



(a) Ambiguous. (b) Ambiguous. (c) Non-ambiguous.

Figure 3.6: Trajectories which are tough and easy to be initialized according to geometric shapes. The red rectangles are start points of each trajectory. The initial trajectories of the first two sequences are lack of geometric characteristics. They are ambiguous ones and cannot be localized. The right most sequence is complex and can be localized.

an exponential model. In fact, as all geometric maps are modelled with the road centre line, there is always an offset between traffic roads and map-represented roads. Besides, the direction of the vehicle's trajectory might not be perfectly aligned with road's direction. As a consequence, the maximum value of v appears when $d_{\mathrm{SM}} > 0$.

3.4.3 Initialization

The initialization step is responsible for the initial state estimation. As we all know, pure odometry based localization system could not possibly give a global position estimation. Here, we rely on the downloaded map and the trajectory shape to provide us with the initial position. Towards this goal, we first let the vehicle moves for a certain distance. A set of uniform distributed cloud drops modelling scale ambiguity and measurement randomness is generated over all the possible locations. Then the historical trajectory of each drop is converted to a query edge map, which is matched with the template edge map through shape matching. A set of candidate drops with high matching probability is selected and used to estimate the initial position.

As shape matching performance heavily relies on the geometric complexity of the trajectory, whether the initialization process can succeed or not depends on practical road conditions at the initial stage. Initial trajectories consisting of short, straight driving (as shown in Fig. 3.6a) or self-similar routes (as shown in Fig. 3.6b) with high inherent ambiguities are less likely to be initialized while trajectories with sufficient complex characteristics (as shown in Fig. 3.6c) are much easier to cope with. The selectivity of trajectory in initialization step is one shortcoming of the proposed method. Similar problem is also discussed in [102], where they call this issue as fundamental ambiguity. In our experiment, to make sure the path is easy to manipulate, the first batch of drops covering all the roads of the map is generated only after the vehicle has moved a minimum distance $d_{\min}$ meters and a minimum turning angle $\alpha_{\min}$.

3.4.4 Drops Resampling and Scale Ambiguity Reduction

After the certainty degree and the consistency are obtained, the weight of each cloud drop $\mathbf{y}_i$ is determined by the product of its certain degree $\mu(\mathbf{y}_i)$ and *consistency*

$\upsilon(\mathbf{y}_i)$:

$$w(\mathbf{y}_i) = \mu(\mathbf{y}_i)\upsilon(\mathbf{y}_i). \tag{3.27}$$

The weights of cloud drops are positively correlated to both certainty degree and constraint consistency. To reject cloud drops that are with lower probability to be the correct location, the robust low variance resampling algorithm is used. The cloud drop set $\mathcal{S}$ is then acquired by taking all cloud drops after resampling.

As we talked before, the existence of scale drift makes the absolute scale of monocular visual odometry varies as the vehicle moves. A parameter estimation scheme is desired to narrow down the scale ambiguity while resampling drops. Based on the analysis of Section 3.3.4 and Property 2-3, with the method of moments applied, the parameters can be estimated based on the following underdetermined system:

$$M_{1j} = \hat{E}(S_j)\hat{E}(X_j) \tag{3.28}$$

$$M_{2j} = \hat{\sigma}_{X_j}^2 \hat{E}(S_j)^2 + \hat{\sigma}_{S_j}^2 \hat{E}(X_j)^2 + \hat{\sigma}_{S_j}^2 \hat{\sigma}_{X_j}^2, \tag{3.29}$$

where $M_{1j} = \frac{1}{N}\sum_{i=1}^{N} y_{ij}$ and $M_{2j} = \frac{1}{N-1}\sum_{i=1}^{N}(y_{ij} - M_{1j})^2$ denote the expectation and variance of the j-th random variable in a cloud drop, respectively; N is the total number of cloud drops. Obviously, the above underdetermined system have fewer equations than unknowns and it is difficult to solve.

As the raw measurement is quite accurate over a short distance, the scaling vector plays a major role in selecting cloud drops. To simplify the problem, we assume that road constraints filter out cloud drops with improper scaling factors. Alternatively, the following assumption can be made:

Assumption 3. *The parameters related to raw measurement are identical before and after cloud drop selection. Mathematically,* $\hat{E}(X_j) = E(X_j), \hat{\sigma}_{X_j} = \sigma_{X_j}$.

In real MVO, the elements in the scaling vector $\mathbf{S}$ are identical for all directions. Besides, only 2D road constraint is provided from OpenStreetMap. Thus, the pa-

rameter estimation equations lead to an overdetermined system

$$M_{1j} = \hat{E}(S_j)E(X_j) \tag{3.30}$$

$$M_{2j} = \sigma_{X_j}^2 \hat{E}(S_j)^2 + \hat{\sigma}_{S_j}^2 E(X_j)^2 + \hat{\sigma}_{S_j}^2 \sigma_{X_j}^2, \tag{3.31}$$

which is rewritten as $\mathbf{A}\hat{\beta} = \mathbf{M}$:

$$\begin{bmatrix} E(X_1)^2 & 0 \\ E(X_2)^2 & 0 \\ \sigma_{X_1}^2 & E(X_1)^2 + \sigma_{X_1}^2 \\ \sigma_{X_2}^2 & E(X_2)^2 + \sigma_{X_2}^2 \end{bmatrix} \begin{bmatrix} \hat{E}(S)^2 \\ \hat{\sigma}_S^2 \end{bmatrix} = \begin{bmatrix} M_{11}^2 \\ M_{12}^2 \\ M_{21} \\ M_{22} \end{bmatrix}, \tag{3.32}$$

whose solution $\hat{\beta}$ can be obtained by using the method of ordinary least squares:

$$\hat{\beta} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{M}. \tag{3.33}$$

Once $\hat{E}(S), \hat{\sigma}_S^2$ are estimated, they can be used as the initial scale parameters when generating new cloud drops in the next iteration.

3.5 Experimental Validation

To evaluate the performance of the proposed approach in real conditions, experiments are conducted on the KITTI visual odometry dataset as well as our self-collected dataset. Both of these two datasets are captured by driving a wagon around urban and suburban environments at normal driving speed. Since many turns are included in these datasets, they are very challenging for pure odometry based localization.

Even though the original intention of this work is to tackle measurement uncertainties of MVO which include the drift and scale ambiguity, localization with both

stereo and monocular visual odometry are conducted, and the results are shown to demonstrate the performance of our proposed approach. The reasons why stereo visual odometry based localization is also evaluated can be explained as follows:

1. Localization with SVO can be regarded as a specific case of MVO when $\mathbf{s} = \mathbf{1}$. Evaluation with a known scale validates the effectiveness of parameter estimation scheme, and is the prerequisite of the experiment with MVO.

2. Although no scale ambiguity exists in stereo case, drift caused by calibration error, feature selection error, matching error, etc. may make the estimated scale fluctuate around the true value. The proposed approach provides a universally applicable framework to deal with these measurement uncertainties.

In experiments, some typical parameters are set as: initialization criterion $d_{\min} = 300$ m, $\alpha_{\min} = \pi/4$; the weight $\alpha = 0.5$ for orientation distance in shape matching; degrees of freedom $k = 3$ for Chi-squared distribution; 800 drops are sampled in the proposed framework. All the experiments are conducted on a mobile workstation with an i7-4710MQ processor.

3.5.1 Experiments on KITTI

The KITTI visual odometry dataset contains 22 stereo sequences, and a part of their ground truth (the first 11 sequences) has been released. These sequences are made with various lengths and trajectory shapes; thus they are suitable for localization performance evaluation. Some of them are used in this experiment.

3.5.1.1 Localization with Stereo Visual Odometry

In the first round of experiment, the proposed approach is evaluated with SVO dead-reckoning results. We choose the widely-used *libviso2* package [105] as SVO

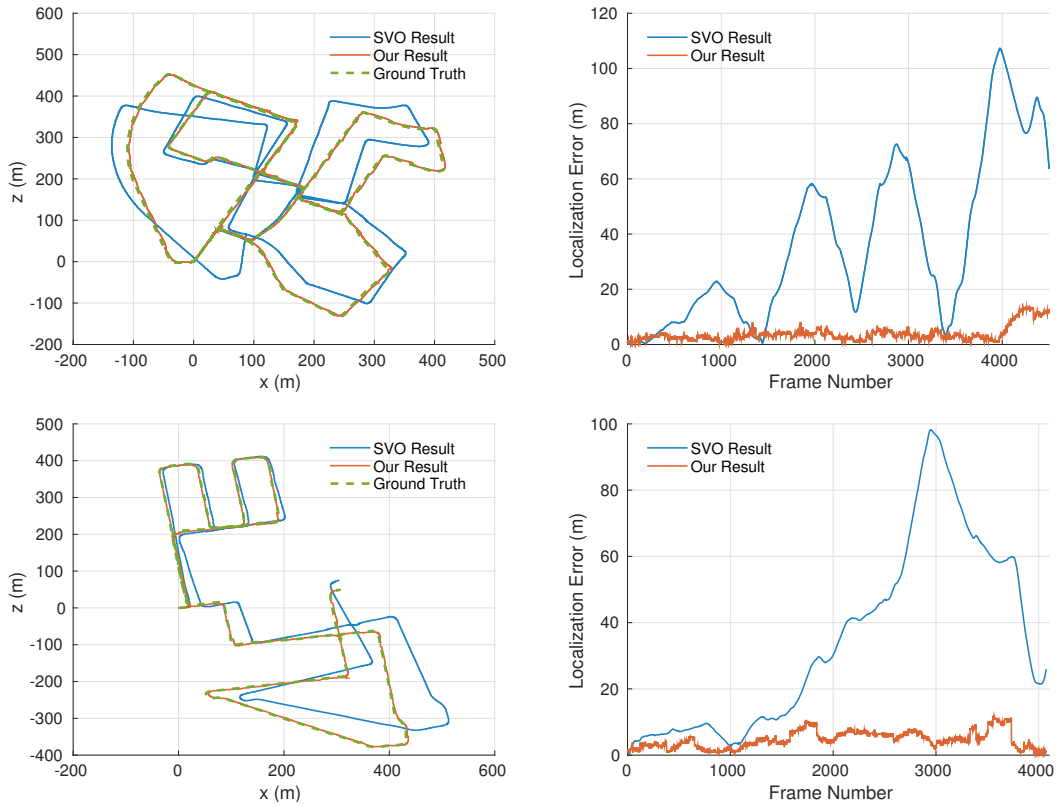


Figure 3.7: Sequence 00 and 08 from visual odometry benchmark of the **KITTI** dataset. Left: trajectory estimated from stereo visual odometry, trajectory estimated from our road-constrained algorithm and ground truth. Right: localization error comparison between SVO and our method.

model. During the experiment, it is assumed that the scale of SVO is unknown, and the initial scaling factor satisfies $\mathbf{S} \sim N(1, 0.5)$. In that sense, the SVO can be regarded as a special MVO.

Some representative trajectories estimated from SVO and our road-constrained approach, as well as the localization error comparison curve can be found from Fig. 3.7. As can be seen, the SVO drift increases every time the robot turns sharply. And due to the accumulative drift, pure SVO localization becomes increasingly unreliable with the growth of travel distance. In contrast, for the proposed approach, the more complicated trajectory is, the better localization results will be. It is not difficult to understand this unique phenomenon, as every time the robot changes its direction of motion, more information is added to robot's trajectory, and the shape matching

Table 3.1: Quantitative comparison between the proposed approach, pure SVO and [102].

Sequence		**KITTI 00**	**KITTI 02**	**KITTI 05**	**KITTI 08**	**KITTI 09**	Total
Travelling Distance (km)		3.72	5.06	2.20	3.21	1.70	10.83
Our Proposed	Avg Error (m)	3.76	11.32	4.02	4.67	5.72	6.59
	Std Dev (m)	2.80	7.51	2.04	2.57	2.67	-
	Max Error (m)	14.01	25.60	8.70	11.96	11.57	-
SVO	Avg Error (m)	37.11	66.08	14.40	34.70	16.63	40.51
	Std Dev (m)	29.63	47.37	13.30	28.12	34.91	-
	Max Error (m)	107.23	172.71	45.60	98.20	9.72	-
[102]	Avg Error (m)	2.1	4.1	2.60	2.4	4.2	3.19

algorithm narrows down the possible region of the robot.

Quantitative results from our method, pure SVO and state of the art map-aided approach **Lost** [102] are listed in Table 3.1. Noticed that not all the 11 sequences of KITTI dataset are listed in the table. That's because some sequences do not satisfy the initialization criterion as discussed in Section 3.4.3. According to the results, both our method and **Lost** perform much better than pure SVO, as expected. The proposed approach suppressed average localization error drastically from 160.07 m to 6.59 m. Besides, the proposed approach provides more robust positioning compared to pure SVO, as the standard deviations for all sequences are evidently lower. Further more, the success of our SVO based localization proves the effectiveness of the proposed framework, especially the effectiveness of our parameter estimation scheme. Finally, even though **Lost** has a slightly better performance than the proposed, the advantage of our approach will be unfolded in monocular case.

Error sources: the road map in our work is converted to an edge map, in which each road segment is represented with one edge. The positioning performance depends heavily on the resolution of this edge map. Theoretically, error from edge map precision can be eliminated if the resolution of the map is high enough. However, this is in contradictory with execution time and memory usage, as shape matching is needed for each possible trajectory to obtain cloud drop consistency. In our implementation, a 0.75 m/pixel (not high resolution) edge map is used. On the other hand, the shape matching algorithm we used in our implementation is a modified

chamfer matching [104]. It works well for straight line segments. For arc segments, even though they can split into small straight lines, it is very challenging to have a reliable results especially when the vehicle makes shape turns. These are the reasons that may cause positioning performance degradation.

3.5.1.2 Localization with Monocular Visual Odometry

In the second round of experiment, we evaluate the proposed approach using the raw translation vectors of MVO without scale information. State of the art monocular SLAM system-ORB-SLAM [32] is used to provide us with monocular motion prediction. Relying on loop closing and pose graph optimization, this system can achieve quite accurate motion estimation and map generation up to an absolute scale. When no loop closure occurs, standard local tracking and mapping step are performed such that ORB-SLAM acts like a VO system. Without distinction, all the raw motion estimations from ORB-SLAM are called MVO in this experiment.

To validate the effectiveness of our proposed method, we compare the performance of our method with MVO method in both conditions with loop closure and without loop closure. Since the absolute scale is unknown, to compare monocular motion estimation with the ground truth, we re-scale the MVO raw measurement by aligning the odometry trajectories during initialization with the ground truth using a similarity transformation, as shown in the left column of Fig. 3.8. As can be seen, MVO performs much better on sequence 00 than on sequence 08 and 09 because loop closures exist in route 00 while they do not exist in the other two sequences (actually, sequence 09 do have loop closures at the very end, but ORB-SLAM fails to detect it). The 7 DOF drifts, especially the scale drift, destroy the motion estimation of sequence 08 and 09. Despite the loop closure optimization, drift issue still exists in sequence 00. By contrast, our road-constrained method always performs well regardless of the existence of the loop closure. Most of the results are perfectly constrained on or near a road thanks to our shape matching and parameter estima-

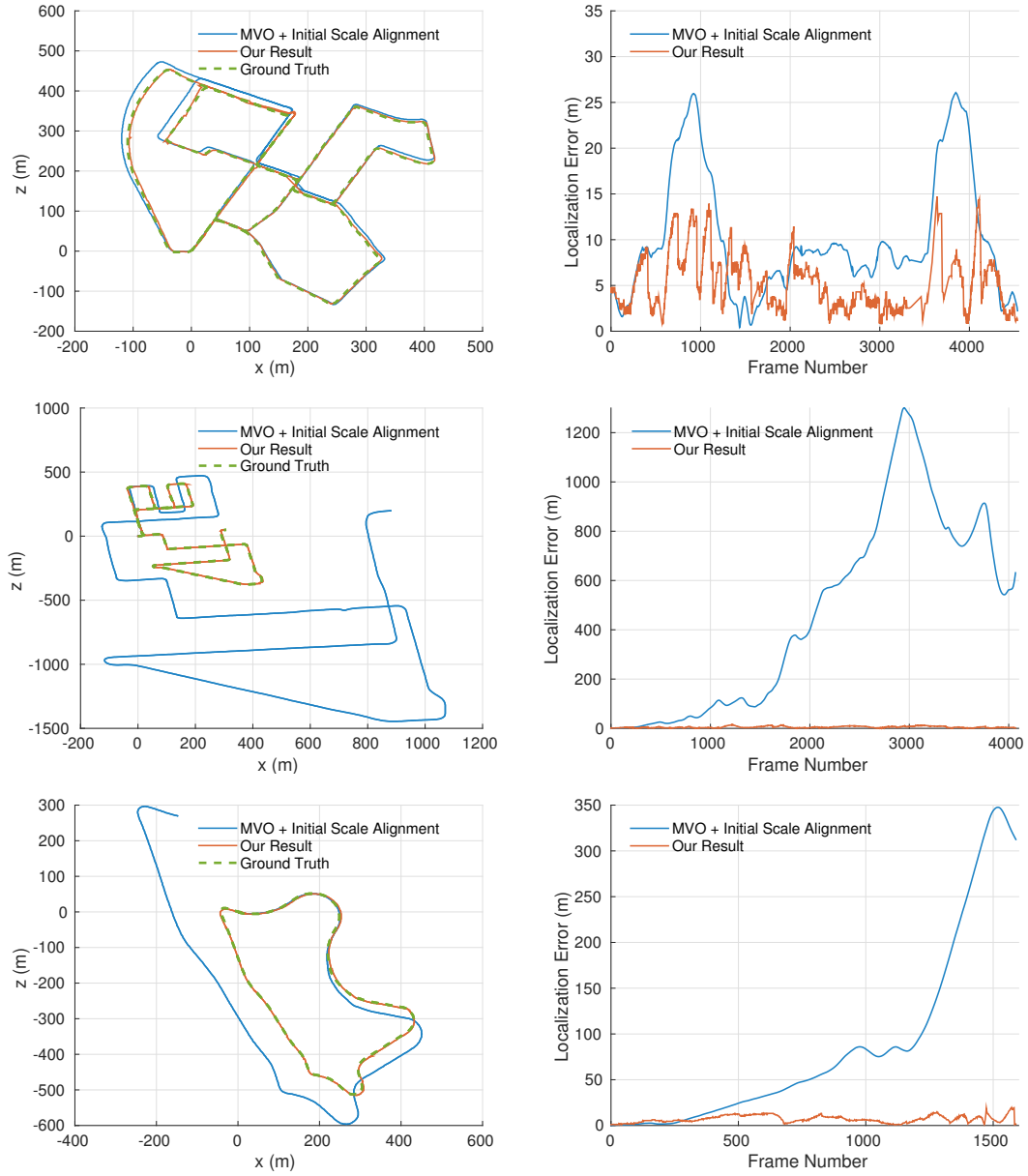


Figure 3.8: Sequence 00, 08 and 09 from visual odometry benchmark of the **KITTI** dataset. Left: trajectory estimated from monocular visual odometry aligned with initial scale, trajectory estimated from our road-constrained algorithm and ground truth. Right: localization error comparison between MVO and our method.

tion scheme. The right column of Fig. 3.8 shows error curves of the two methods. Based on the localization error comparison, it can be concluded that the proposed framework is valid for monocular localization.

The quantitative results are also listed in Tabl 3.2. As KITTI 00 and KITTI 05

have loop closures, MVO results for these sequences are still acceptable. However, MVO results from KITTI 08 and KITTI 09 are not as optimistic as the first two sequences. At the same time, the results from our algorithm are much better, as the average localization error is 5.62 m compared to 160.07 m over the 10.83 km run. Moreover, the proposed approach outperforms **Lost** in this monocular case. As can be seen, our MVO based localization has roughly the same performance as our SVO based, while **Lost** has a larger inconsistency. This furtherer proves the effectiveness of the proposed for both SVO and MVO.

It is worthy to note that, for the monocular positioning case in **Lost** [102], the monocular visual odometry used is not real "monocular". Instead, the metric information from the height and pitch of the camera, together with a detected plane ground is utilized to obtain the completed pose. The translation estimation is at the real scale and no scale ambiguity exists. Normally, odometry result estimated from this method is not as good as stereo case since it requires a good ground plane detection accuracy. That's why [102] performs much worse in "monocular" case than stereo case.

In order to demonstrate the advantage of the proposed cloud model with respect to models which consider measurement randomness and scale ambiguity separately, comparison experiments are made. Concretely, scale ambiguity is firstly modelled while MVO measurement randomness is not considered. In this scenario, the scaled measurement follows $GGC(E\mathbf{S}, \Sigma_S, E\mathbf{X}, \mathbf{0})$. The result is listed as the next-to-last row of Table 3.2. As can be seen, the positioning error of this model is larger than the complete model. It's not hard to understand this result, since measurement randomness is not properly modelled here, translational and rotational drift issues still exist. Experiment evaluating another scenario is also conducted. This time, measurement randomness is modelled while scale ambiguity is not considered. And the scaled measurement follows $GGC(E\mathbf{S}, \mathbf{0}, E\mathbf{X}, \Sigma_X)$ in this case. The result is listed as the last row of Table 3.2. Unsurprisingly, this model performs much worse than the complete model due to the unhandled scale drift issue.

Table 3.2: Quantitative results of the proposed approach, pure MVO aligned with initial scales, [102], model only considering scale ambiguity and model only considering odometry measurement randomness, respectively.

Sequence		**KITTI 00**	**KITTI 05**	**KITTI 08**	**KITTI 09**	Total
Travelling Distance (km)		3.72	2.20	3.21	1.70	10.83
Our Proposed	Avg Error (m)	5.43	6.02	5.05	6.63	5.62
	Std Dev (m)	3.15	4.82	3.23	2.93	-
	Max Error (m)	14.70	18.70	14.67	16.45	-
MVO	Avg Error (m)	10.00	7.21	475.43	90.81	160.07
	Std Dev (m)	6.62	7.20	408.21	100.84	-
	Max Error (m)	26.07	23.31	1300.20	347.64	-
[102]	Avg Error (m)	15.60	5.60	45.20	5.40	16.72
Only Ambiguity	Avg Error (m)	6.01	8.13	66.90	63.40	33.50
Only Randomness	Avg Error (m)	9.20	7.00	462.30	85.30	154.90

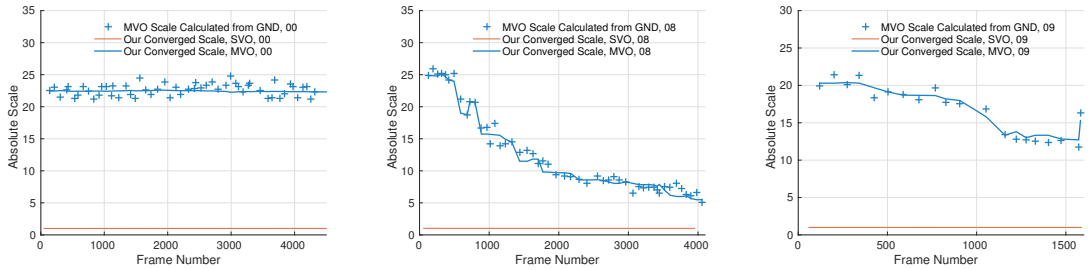


Figure 3.9: The absolute scales of sequence 00, 08 and 09. The red solid lines represent corrected scales of stereo visual odometry. The blue solid lines represent corrected scales of monocular case. Blue crosses show the scales calculated from ground truth.

3.5.1.3 Scale Drift Correction

Fig. 3.9 shows the absolute scale estimated from the proposed method. Red and solid blue line are the estimated scales of stereo and monocular odometry respectively. The "true" scale at a certain distance—represented with blue cross, is calculated from the ratio between ground truth and the raw odometry measurement. As can be seen, stereo odometry and monocular odometry performs differently and different sequences also perform diversely. Most of the stereo ones are successfully converged to the ideal scale—one, which is precisely what we need to verify the feasibility of the proposed parameter updating scheme. For the monocular case, sequence 00 has a nearly constant scale, while sequence 08 and 09 have greatly drifted scales. Even though the converged scales are not exactly the same with the scales calculated

from ground truth, our results fit the changing tendency well. Please note the huge variation of the scales of 08 and 09, which further verifies the effectiveness of our parameter estimation part.

3.5.2 Experiments on Self-Collected Dataset

Figure 3.10: Our evaluation vehicle.

In this section, experiments conducted on our self-collected dataset are described. Fig. 3.10 shows our evaluation vehicle and the sensor setup. A stereo camera set is mounted on the roof of the vehicle and oriented forward. It is configured to acquire stereo frames at 50 Hz with a resolution of 1280 x 1024. The vehicle is also equipped with GPS to provide us with position ground truth. Our datasets are captured by driving around the campus of Nanyang Technological University.

The right column of Fig. 3.11 shows the OpenStreetMap used in our experiment. The pink trajectory at the bottom left corner shows the route of our vehicle. It is a 1.2km closed loop path around hall seven of NTU.

Similar to the first experiment on KITTI dataset, both stereo and monocular case are considered here and same visual odometry packages are utilized. Trajectories estimated from our method, as well as trajectories estimated from pure stereo visual odometry and monocular odometry systems are shown in Fig. 3.11 and 3.12. Unsurprisingly, poor results are obtained from pure odometry systems. Moreover, stereo odometry performs even worse than monocular one since no loop closing technique

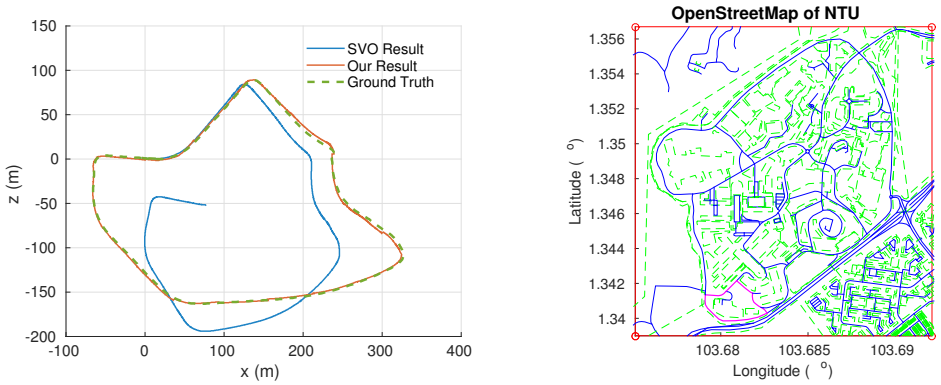


Figure 3.11: Our self-collected dataset. Left: trajectory estimated from stereo visual odometry, trajectory estimated from our road-constrained algorithm and ground truth. Right: the map used in our experiment; the pink trajectory shows the route of our vehicle.

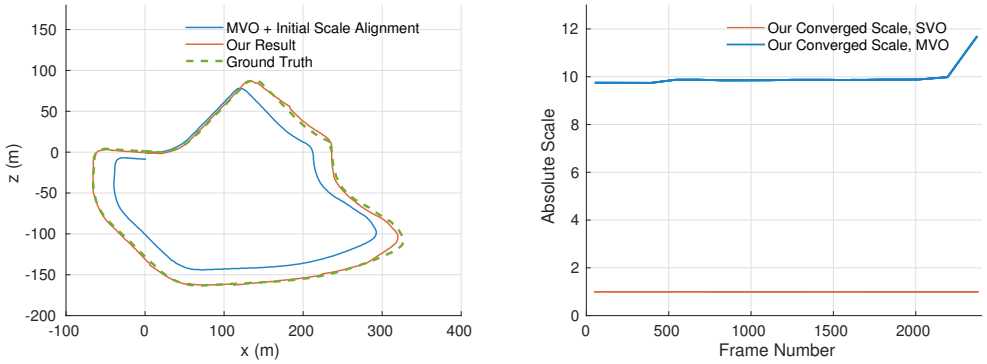


Figure 3.12: Our self-collected dataset. Left: trajectory estimated from monocular visual odometry aligned with initial scale, trajectory estimated from our road-constrained algorithm and ground truth. Right: scales estimated from our method.

is used. At the mean time, our proposed method performs very well as always. The right most column of Fig. 3.12 shows the absolute scale estimated from our method. Obviously, the stereo case can still converge to one correctly. While for the monocular case, two sudden changes of scale are occurred around frame 500 and 2200, when the vehicle happened to make turns.

Experiments analysing the importance of parameter setting are also conducted. The two curves of Fig. 3.13 demonstrate the influence of the given parameters (ES, σ_s) on the convergence performance. As can be seen from the left curve, the estimated scale can converge to the truth when the given scale's expectation lies in the rough

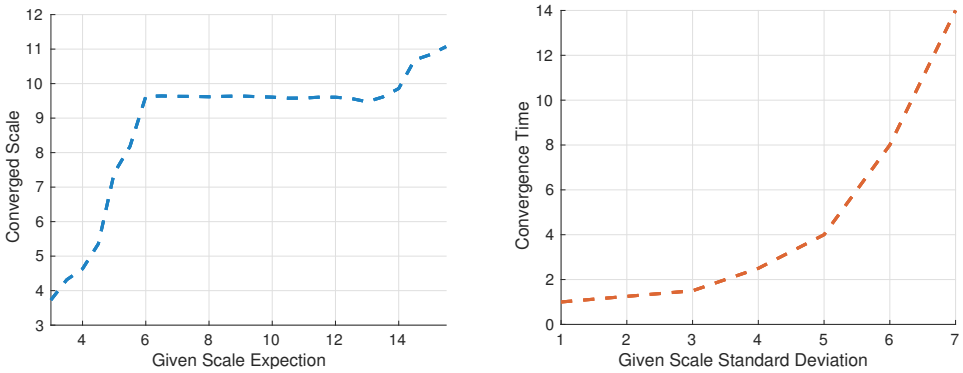


Figure 3.13: The influence of the given parameters. Left: the influence of given scale exception ES on final converged scale; the true scale is 9.7, $\sigma_s = 3.5$. Right: the influence of given scale standard deviation σ_s on the convergence time; the smallest time consumption is considered as one time unit.

range $[ES_{truth} - \sigma_s, ES_{truth} + \sigma_s]$. It will be always possible to converge to the true scale as long as the given standard deviation is large enough. Nevertheless, as shown from the right curve of Fig. 3.13, the convergence time used to converge to the true position increases exponentially over the given standard deviation, which verifies the significance to narrow down the scale ambiguity while resampling drops.

The most time consuming step of the proposed framework is the road constrained shape matching process. It is worthy to note that this step is not conducted at every time instant, but when the local history has a length of $l = 80$ frames. The run-time of this step is mainly determined by the size of the map. With map size 1 km^2, it takes 1.06 ms per drop for shape matching. The evaluation of other steps in the pipeline takes less than 0.01 s.

3.6 Conclusions

In this chapter, we have presented a localization framework to globally localize a mobile vehicle equipped with monocular visual odometry and the freely available OpenStreetMap. In this framework, the Gaussian-Gaussian cloud model has been proposed to represent measurement randomness and scale ambiguity from raw mea-

surement of visual odometry. A shape matching scheme has been used to filter out cloud drops which are inconsistent with road constraints. With the integration of shape matching, the vehicle can be robustly localized while the measurement uncertainties are reduced. A parameter estimation scheme has been implemented to narrow down the scale ambiguity while resampling cloud drops. Experiments on the widely used KITTI benchmark and our self-collected dataset have shown that the proposed approach is effective for both stereo and monocular odometry, and the localization error of the proposed approach could be significantly reduced compared to pure SVO and MVO based methods.

Chapter 4

Place Recognition Using Multiple Feature Types

4.1 Introduction

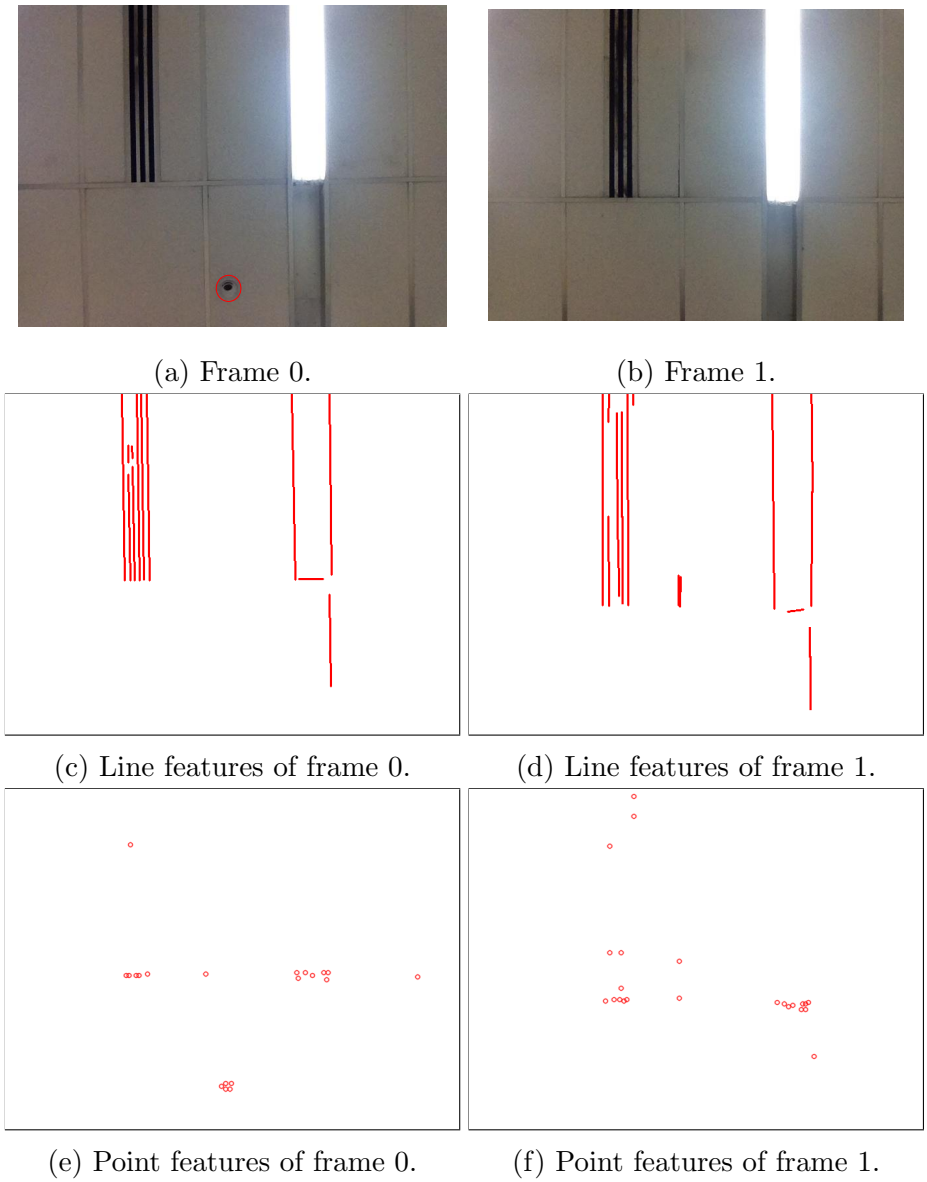
In previous chapter, we focus on the development of metric localization based on road-constrained visual odometry. In this chapter, efforts are given to robust place recognition.

Vision-based place recognition problems can be considered as that, given a query image captured at a particular place, return images that depict the same place from geo-tagged database. Efficient place recognition methods often build on bag-of-words (BOW) representation developed for object and image retrieval. There are three steps: feature extraction, vocabulary construction and vector quantization to represent an image with bag of visual words. In the first step, for each database image, its local features are detected and described with invariant descriptors. A k-means clustering algorithm is usually applied in the second step to construct visual words vocabulary and the resulting cluster centers (i.e., centroids) are treated as dictionary of visual words. Given one query image, all the local features are

extracted first and the corresponding descriptors are quantized to find the nearest neighbour in the dictionary created in the second step. The histogram of all the visual words forms the final bag-of-words vector.

Point features, such as SIFT [78] and rotated BRIEF [85], are widely used to represent a scene in current place recognition systems. One of the most well-known approach is Fast Appearance-Based Mapping (FAB-MAP) [81], which can perform very large trajectory estimation based on bag of visual words model using SIFT features. Besides point features, there are other popular types of features such as line features that are used in different applications. Compared with point features, line features carries more structural information since each of them is spanned over a 2D space instead of a single point. Moreover, they are more robust to environmental changes such as illumination, viewing direction, or occlusion. Line feature based place recognition approach is expected to have a better performance in well structured environments. However, there is no such feature that outperforms others in all environments. Each feature has its own advantage, they should be carefully chosen depending on the context and environments.

In [106], my colleagues use lines extracted from ceiling images to calculate the robot motion and use ceiling as an absolute reference to determine the robot's global orientation. Although the motion estimation approach demonstrates robustness on disturbance, place recognition using ceiling vision can be even more challenging due to the lack of point features and line patterns high similarity. As can be seen in Fig. 4.1, lines are the dominant patterns in both images. Since they are similar in the two images, it is difficult to distinguish two places using line features alone. On the other hand, the number of point features is too small (about 20 FAST features in each image) to generate a meaningful set of visual words to describe an image properly. Nevertheless, it is worthy to note that point features can provide extra distinctiveness to find correct matchings in a database. Thus, a good combination of both kind of features is supposed to have a better performance on place recognition.



(a) Frame 0. (b) Frame 1.

(c) Line features of frame 0. (d) Line features of frame 1.

(e) Point features of frame 0. (f) Point features of frame 1.

Figure 4.1: Point and line features extracted from ceiling images. Frame 0 and frame 1 looks very similar from line features, but these two frames were taken at different locations (note the fire alarm in frame 0). In order to distinguish these frames, we combine line and point features together by using a modified vocabulary tree.

In this chapter, we present a place recognition system which is applicable to various environments. The main constructions of this system are as follows:

1. A modified vocabulary tree with the ability of merging multiple kinds of features is proposed. Although this method is initially proposed for ceiling vision

based place recognition with the combination of point and line features, users can customize different combination of features for different environments.

2. Instead of building separate vocabulary trees for each feature type, one single vocabulary tree is created for all feature types, which makes the proposed much easier to work with.

3. Validation experiments are conducted on real-world datasets and experimental results have demonstrated the advantage of our system compared to existing approaches.

In this study, we focus on the combination of point and line features. The rest of this chapter is structured as follows: In Section 4.2, we describe how to extract and describe point and line features. After that the modified vocabulary tree that can combine multi features types is introduced in Section 4.3. Experiments conducted to evaluate the proposed approach are described in Section 4.4. Conclusions are drawn in Section 4.5.

4.2 Point and Line Extraction

We discuss how point and line features are extracted and described in this section. For ground robots using ceiling vision in a typical indoor environment, the scale of different images remain constant but not the orientation of features. Hence, a feature descriptor that is invariant to the rotation is desired. To maximize the efficiency of the system, a specially designed point feature descriptor is used. As for line features, we use Line Band Descriptor (LBD) [107] to represent lines due to its high reliability and computational efficiency.

4.2.1 Point Extraction and Description

A large number of point feature detection approaches have existed in the literature. SIFT and SURF are very popular choices for visual feature detectors for their robustness against lighting and scale changes. However, without the help of modern GPU, it is time consuming to detect SIFT or SURF features, which does not meet our purpose for real-time performance. Hence, we use FAST [84] feature detector with non-maximal suppression. In order to increase the processing speed and reduce the number of features that are very close to each other, the images are down-sampled before applying feature detection and transform these feature locations back to full-size images when performing descriptor calculation.

Feature descriptors are extracted to find feature similarity across frames. Because we intend to speed up place recognition process, scale invariant feature descriptors such as SIFT or SURF are computational expensive and in our ceiling vision application scale invariance is not a necessary requirement of the system. To fulfill the real-time requirement, a customized and fast feature descriptor is desired.

For each frame, we first apply two 5×5 Sobel filters (Fig. 4.2a and 4.2b) across the down-sampled image along horizontal and vertical directions. For each point in the image, G_x and G_y are the resulting Sobel responses along horizontal and vertical directions. The feature direction can be obtained as $atan2(G_y, G_x)$ and the gradient magnitude can be represented as $G = \sqrt{G_x^2 + G_y^2}$. The descriptor is designed as concatenation of gradient magnitudes around detected features.

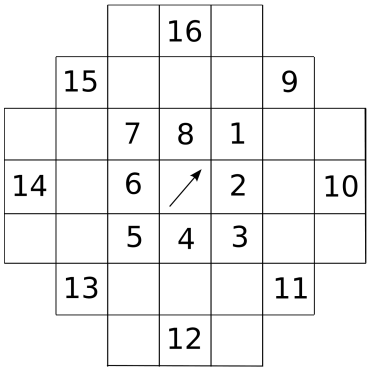
Its layout consists of two circles with 16 coding positions as shown in Fig. 4.2c. The descriptor is encoded sequentially from inner circle to outer circle and both follow a clockwise direction. The feature direction is quantized into 8 discrete directions which determine the starting coding position. The value for each dimension of the descriptor is scaled into $[0, 255]$ so that each descriptor can be represented by 16 bytes. By this simple design, the descriptor can be extracted efficiently and

1	2	0	-2	-1
4	8	0	-8	-4
6	12	0	-12	-6
4	8	0	-8	-4
1	2	0	-2	-1

(a) Horizontal Sobel kernel.

-1	-4	-6	-4	-1
-2	-8	-12	-8	-2
0	0	0	0	0
2	8	12	8	2
1	4	6	4	1

(b) Vertical Sobel kernel.



(c) Descriptor layout.

Figure 4.2: Sobel kernel and descriptor layout.

invariant to rotation and illumination changes. The descriptor similarity is defined as sum of absolute difference (SAD). In order to optimize the implementation, we use the *Streaming SIMD Extensions* (SSE) instructions which can perform 16-byte operations very efficiently. Hence, the 16-byte SAD can be performed within one call of SSE instruction.

4.2.2 Line Extraction and Description

In order to extract line segments from input images, we use a method presented in [107]. The extraction starts with a scale-space pyramid consisting of N octave images which are generated by down-sampling the original image with a set of scale factors and Gaussian blurring. Then, from each layer in the pyramid, EDLine [108] algorithm is applied, which produce a set of lines in the scale space. Lastly, all the detected lines are re-organized by finding correspondences in the scale space.

After the extraction of all the line segments, a binary descriptor representing each line is generated using LBD. Given a line segment, a local rectangular region around the line is chosen as the line support region (LSR). The LSR is divided into a set of bands $\{B_1, B_2, ..., B_m\}$, whose length equals the one of line. The line direction d_L together with the orthogonal direction $d_\perp$ forms a local 2D reference frame, whose origin is located at the middle of the line. The gradient of each pixel in the LSR is

projected to the local frame. Later on, a global Gaussian function f_g is applied to all rows in the LSR, while a local Gaussian function f_l is applied to all rows in each band and its nearest two neighbor bands. Using the local and global Gaussian function as well as the gradient of each pixel in the LSR, a so called band description matrix (BDM) is constructed. Each band B_j in LSR has an assoicated band descriptor (BD), which can be obtained using the mean and standard deviation of the BDM. Once each band has been assigned its BD, the Line Band Descriptor LBD is simply generated by concatenating them:

$$LBD = (BD_1^T, BD_2^T, ..., BD_m^T). \tag{4.1}$$

4.3 Modified Vocabulary Tree

The vocabulary tree based method is a widely used one for place representation [83]. It is built by hierarchical k-means clustering, which makes vocabulary tree approach a practical unsupervised learning method. In this section, we discuss how to create conventional vocabulary tree first and then introduce how to create our modified vocabulary tree. Finally, we give the explanation on how to create the database with the modified tree.

4.3.1 Conventional Vocabulary Tree Creation

In conventional vocabulary tree approach, in order to build the tree structure, a large number of features are extracted from some training images first. Then, an initial k-means process runs on the training descriptors, obtaining k cluster centers, which form the first level of nodes in the vocabulary tree. The same process is then applied to the subsequent levels (up to L levels). Finally, a tree with a maximum number of k^L visual words is constructed. Given the constructed vocabulary tree, a BOW vector can be computed for each database image. And all these BOW vectors

form a database for query purpose.

Although this method has been demonstrated to be scalable and efficient [86], it can not be used for applications with multiple feature types directly due to the following reasons. First of all, different feature types may have different dimensions, e.g. SIFT descriptors have 128 dimensions while SURF descriptors have 64 dimensions, which makes it impossible to compare SIFT and SURF descriptors. It is also problematic for different feature types with the same dimension. For example, a 32-dimension ORB descriptor vector could be very close to a 32-dimension LBD descriptor vector in Euclidean space. In this circumstance, the two features will be clustered into the same visual word. Obviously, this is not acceptable since ORB and LBD descriptor convey completely different visual meanings.

It could be argued that multiple separate vocabulary trees can be built for different feature types to solve the above issues. However, in that case, not only several vocabulary trees but also multiple databases are required for different feature types. Besides, when comparing the query image with an database image, multiple scores are returned for different feature types. As we know, different features perform differently in different environments (e.g. line features have better performance in environment like corridor but not in rural area filled with trees or grass). Hence, these scores should not be treated equally in different environments. But it is unclear on how to find a general weighting strategy to combine these scores to perform well in different environments.

4.3.2 Modified Vocabulary Tree Creation

Based on the above observations, we propose a modified vocabulary tree as depicted in Fig. 4.3. As can be seen, a feature type quantization step is performed on the training data, defining n (there are 2 in Fig. 4.3) feature types. The training descriptors are then partitioned into n groups, where each group consists of the descriptor

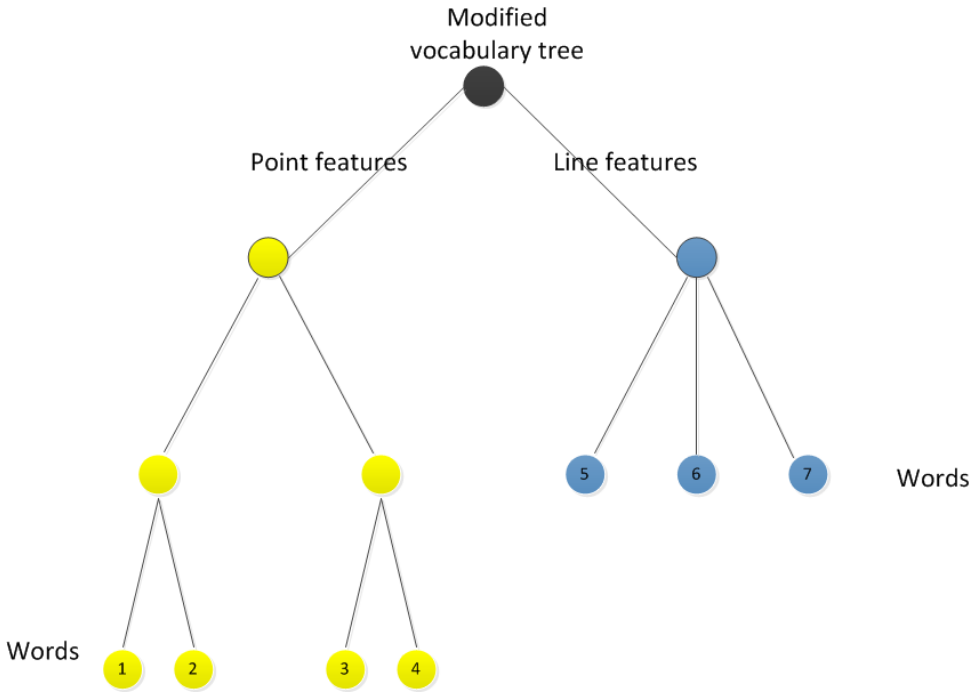


Figure 4.3: Example of our modified vocabulary tree with a combination of point and line features. Instead of running k-means clustering on the first level, a feature type quantization is performed first. In this example, there are 3 levels and 2 branches on each level for point features and 2 layers with a branch factor 3 for line features.

vectors that belong to the same feature type. The hierarchical k-means clustering process is then applied to each feature type, constructing the final tree structure.

In order to distinguish different feature types in the second layer of vocabulary tree, one extra byte is appended to the end of each feature descriptor to represent feature type in our implementation. It is worthy to note that this tree structure can allow us assign different branch and layer factors for different feature types.

One could argue that there are more than one trees for the heuristic mixing of features and can not construct a unique modified vocabulary tree. However, the authors of [86] have done plenty of experiments comparing different parameter settings (branch and layer factors) and suggested appropriate ones. Even though there are more than one tree for the heuristic mixing of features, the one with the most appropriate setting is preferred. The modified vocabulary tree which combines multiple feature types can be construed as long as the conventional trees trained with sepa-

rate feature type can be constructed. It has all the advantages that the conventional trees have.

4.3.3 Database Creation

To do place recognition, an image database need to be pre-built. When adding an image $\mathbf{I_t}$ to the database, multiple types of features are extracted from the image and each descriptor vector traverses through the vocabulary tree until it reaches any leaf node i. $\mathbf{I_t}$ is inserted to an image list stored in ith node so that when querying an image with the database, comparison is only performed with those images that have common words. To accomplish this efficient comparison, the term frequency-inverted document frequency (TF-IDF) scheme is used. The ith element of query vector $\mathbf{q}$ and database image vector $\mathbf{d}$ are defined as:

$$q_i = n_i w_i, \tag{4.2}$$

$$d_i = m_i w_i, \tag{4.3}$$

where n_i is the term frequency of word i in the query image while m_i is the term frequency of word i in the database image which are defined as:

$$n_i = \frac{Q_i}{Q}, \tag{4.4}$$

$$m_i = \frac{D_i}{D}, \tag{4.5}$$

where Q_i is the number of word i in query image; Q is the number of total words in query image; D_i is the number of word i in database image; D is the number of total words in database image. The reason why we choose to use the number of total words instead of using the number of words that belong to one feature type in query and database images, is to reach fairness between feature types with different feature numbers.

Within this TF-IDF scheme, w_i is defined as:

$$w_i = ln\frac{N}{N_i}, \tag{4.6}$$

where N is the number of images in the database, and N_i is the number of images in the database with word i. By using this weighting scheme, the frequently occurred words are penalized, while the words that are more distinctive are awarded.

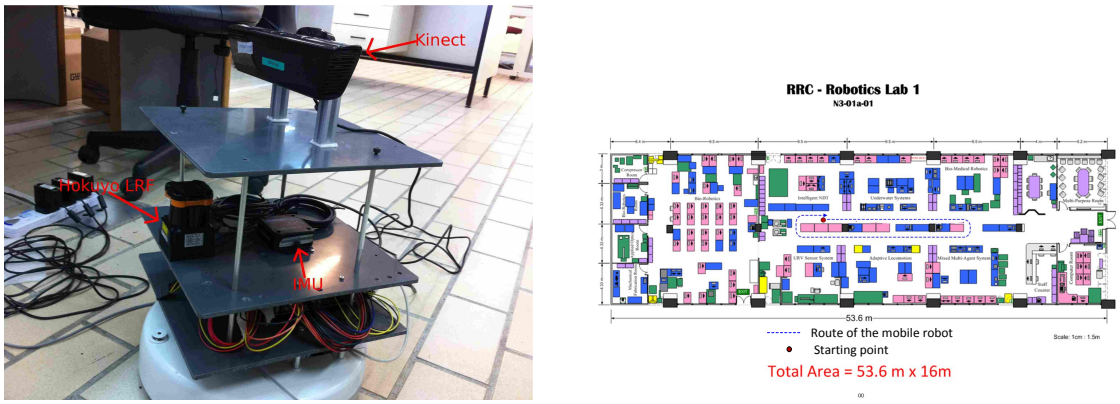
As a result, images are described by bag-of-words vectors. The normalized L_1 distance $dist(\mathbf{q}, \mathbf{d})$ between two such vectors $\mathbf{q}$ and $\mathbf{d}$ is defined as:

$$dist(\mathbf{q}, \mathbf{d}) = |\frac{\mathbf{q}}{|\mathbf{q}|} - \frac{\mathbf{d}}{|\mathbf{d}|}|. \tag{4.7}$$

For a query image, the better matches in database images receive lower $dist(\mathbf{q}, \mathbf{d})$.

4.4 Experiment Validation

In this section, several experiments are conducted and the results are shown to demonstrate the performance of our proposed approach. In the first experiment, we use a Kincect camera pointing up to the ceiling to grab image sequences. Comparisons are made between the performance of our modified tree trained with a combination of LBD line descriptor and point descriptor, and other trees trained using LBD and point descriptor separately. The proposed method is also tested under varied illumination conditions using ceiling images. Afterwards, the challenging KITTI dataset is used in our second experiment to show the effectiveness of the proposed approach in much more complex environment. All the experiments are conducted in real-time using a Intel (R) i7-4710MQ processor.



(a) Our iRobot Create plateform. (b) Robot trajectory.

Figure 4.4: Our mobile robot and its trajectory. The Kinect camera is mounted on the iRobot Create platform and pointed up to the ceiling. The robot traversed the same route three times at different time of a day. Using these three image sequences, we compare the retrieval performance of the proposed modified vocabulary tree trained with both point and line features with conventional vocabulary trees trained with point and line feature alone.

4.4.1 Experiments on Real Robot with Ceiling Camera

The ceiling images used in this experiment are captured using a Kinect mounted on a iRobot-Create robot (as shown in Fig. 4.4a) pointing up to ceiling. The robot is manually navigated by a wireless game pad and the testing data are collected by driving the robot in the Robotics Research Center of Nanyang Technological University (as shown in Fig. 4.4b). The platform is also equipped with a high precision IMU and Hokuyo laser range finder which are used to provide the position ground truth of the robot.

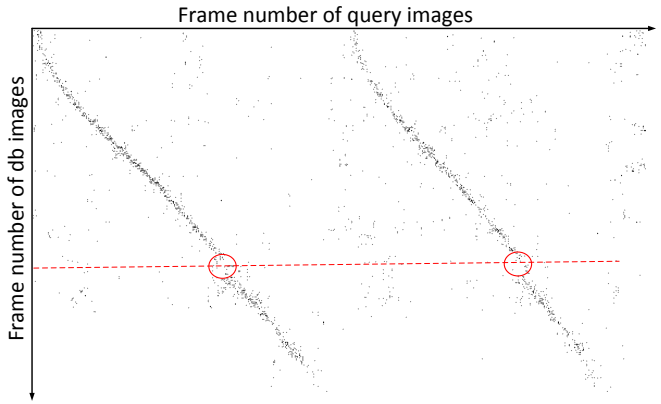
In order to show the effectiveness of the modified vocabulary tree, comparisons between conventional vocabulary trees and the proposed are performed. We first let the robot move around the research center and recording a ceiling video. The proposed point descriptor and LBD line descriptor are used to describe point and line features, receptively. Around 2 million point features and 1 million line features extracted from this video are used to train the conventional and modified vocabulary trees. Following the analysis of [86], a configuration with $k = 8$, $L = 5$ is used for

the line vocabulary tree. And 20200 visual words are generated. While for the point vocabulary, $k = 7$, $L = 5$ is used, resulting in 8987 visual words. The same setting is applied to our modified vocabulary tree.

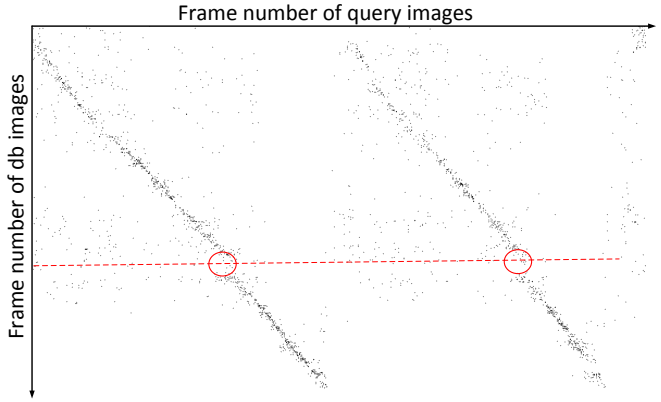
After creating the above vocabulary trees, we first let the robot travel the same trajectory three times but at different time of a day. The blue dashed line in Fig. 4.4b is the route used in this experiment. The first loop was collected in the morning when the lights were still on, while the second loop was collected at noon when the lights were off, and the third loop was collected at night when the lights were on again. So the scenes of the three loops were under illumination changes. We choose the images (1104 frames) in the first loop as database while images in the second (1112 frames) and third (1091 frames) loop are used as query. Thus, images in the second and third loop are matched with images in the first loop sequentially.

The three vocabulary trees are tested using the same ceiling image sequence. In order to show the place recognition performance, we plot the top five query results as shown in Fig. 4.5. The horizontal and vertical axes represent the frame numbers of the database and query images, respectively. The dots represent the top five results returned from database. Since the robot traversed the same trajectory when collecting the three sequences, the distribution of plotted points should follow two diagonal lines. It can be seen, the result from the proposed approach outperforms the ones using single feature type since more dots are distributed on the two diagonal lines. There are some obvious query failures as marked by red circles which occur at the same position of the database images. The reason for that is, there is slight trajectory difference between query and database images when robot made a turn.

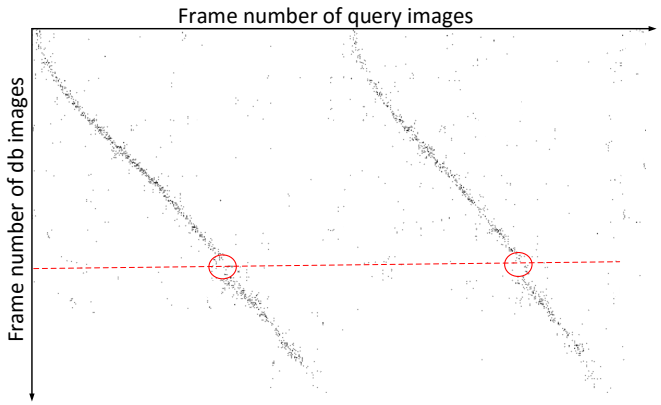
The detailed comparison results of the three vocabulary trees are shown in Table 4.1. From the table it can be seen that the results from both query sequences using the modified vocabulary tree have higher success retrieval rate even under different illumination conditions. Some examples of successful retrieval under illumination changes using the proposed method are shown in Fig. 4.6.



(a) Query result using point features.



(b) Query result using line features.



(c) Query result using both point and line features.

Figure 4.5: Query results using separate vocabulary trees and our modified vocabulary tree that combines point and line features. Two axes represent the frame numbers of the database and query images. The dots represent the top five results returned from database.

Table 4.1: Experimental evaluation of the trees, showing the retrieval performance improvement compared to conventional vocabulary trees. The success return is the number of success retrievals and success rate is the successful retrievals in percentage.

		DB: First Loop (morning, lights on)	
		Success Return	Success Rate
Q: Second Loop (noon, lights off)	Tree Trained with Point Features	763/1112	68.61%
	Tree Trained with Line Features	713/1112	64.12%
	Modified Tree	872/1112	78.42%
Q: Third Loop (night, lights on)	Tree Trained with Point Features	741/1091	67.91%
	Tree Trained with Line Features	701/1091	64.25%
	Modified Tree	858/1091	78.64%

4.4.2 Experiments on KITTI Dataset

In order to quantitatively evaluate the proposed method and demonstrate its ability in much more complex environment, we use the challenging KITTI dataset which contains both urban and sub-urban environment. The position ground-truth of each captured frame in KITTI dataset is available. We use image sequences with number 00, 02, 05 and 06, in which loop closures occur most frequently, so that we can use images in the first pass as database images and images in the second pass as query images. The modified vocabulary tree combining point and line features is created in a similar way as discussed above. One difference is that we use a setting of $k = 10$, $L = 6$ to create the proposed vocabulary tree, which leads to 739725 point visual words and 349613 line visual words.

The database and query image numbers for each sequence are shown in Table 4.2. We compare our method with the popular FAB-MAP method. The precision-recall curves for sequence 00, 02, 05 and 06 are shown in Fig. 4.7. Here, precision is the proportion of returned frame pairs that are correct, recall is the proportion of total correct frame pairs that are returned. It can be seen that our approach shows significant improvement on both the precision and recall rates. In Fig. 4.8, we show examples of correct loop detections in the four KITTI sequences, with point and line corresponding features. Note that the feature matching example image is taken in a well-structured environment which has enough saliency. But pure point and

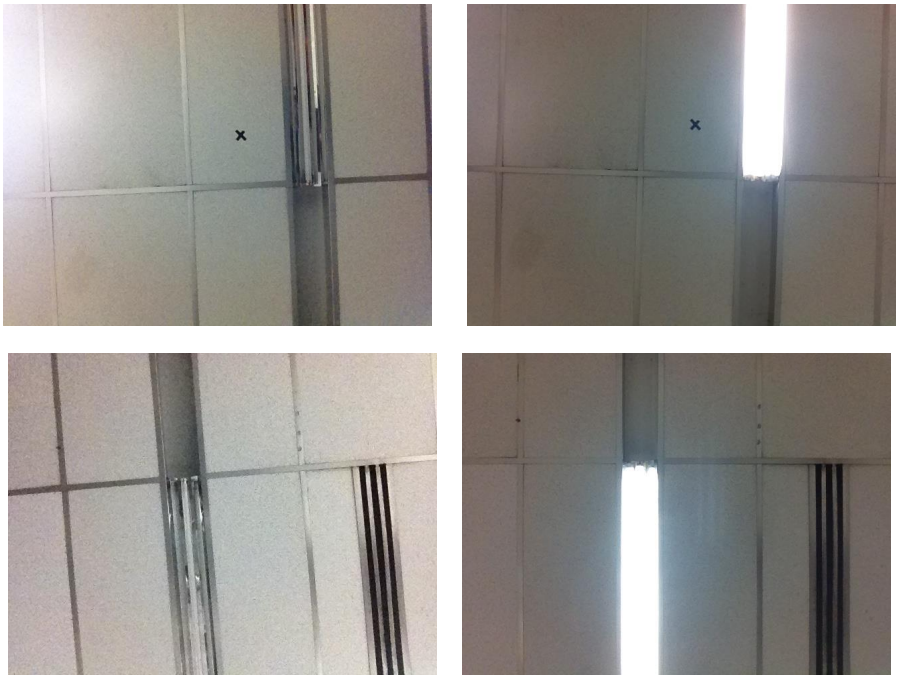


Figure 4.6: Examples of successful retrieval under illumination changes using the proposed method. Left images are queried scenes in the second loop and right images are the returned scenes from database with highest scores.

Table 4.2: Frame numbers of database and query images in the four KITTI sequences.

Sequence No.	Total No. of Frames	Database images	Query images
00	4541	0 - 1570	3400 - 3900
02	4661	0 - 4000	4400 - 4600
05	2761	0 - 1200	1300 - 1600
06	1101	0 - 800	830 - 1090

line feature matching results still have a certain percentage of mis-matches.

Since image position ground-truth is available, given a query image, we measure the Euclidean position distance between the query image and its best matching in the database to evaluate the accuracy of the proposed system. The results are shown in Table 4.3. Different distance threshold levels are defined. One query is considered as correct if the measured distance is smaller than or equal to the distance threshold. The smallest distance threshold is set to one meter, because the average distance between two consecutive frames is about one meter in KITTI dataset. By setting

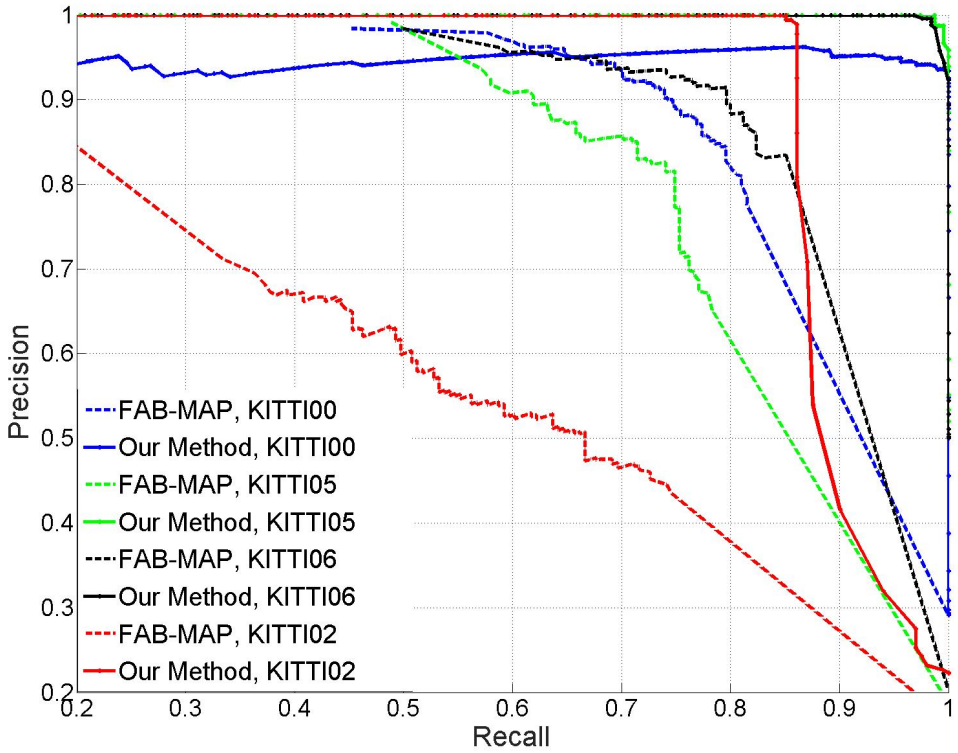
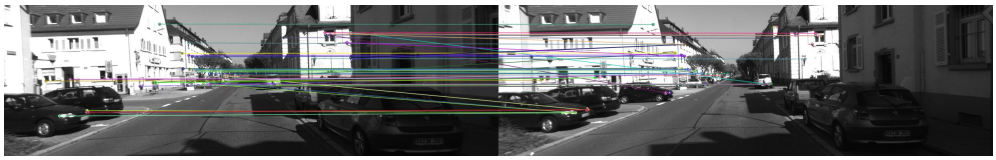


Figure 4.7: Precision-recall curves on four KITTI dataset sequences without geometrical or temporally consistent check, showing the performance improvement compared to FAB-MAP.

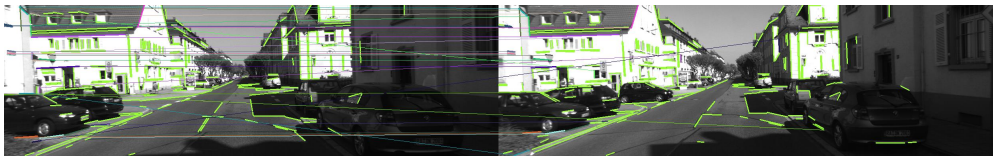
the distance threshold to one meter, a query result is considered as correct only if it is taken at the same position as query image. Most of the query results are correct even with this smallest distance threshold except sequence 02. The reason for that is the car was driving through an area filled with trees and grass which is less distinctive than other sequences.

Table 4.3: Quantitative evaluation of query performances. **Dist. Threshold**: distance threshold within which the query results are considered as correct. **Avg. Error**: average Euclidean distance between the positions of query and result images.

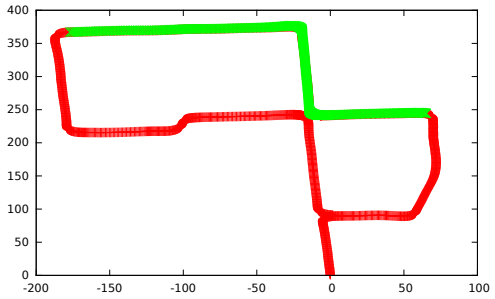
Dist. Threshold	**KITTI00**		**KITTI02**		**KITTI05**		**KITTI06**	
	true	false	true	false	true	false	true	false
$\leq$ 1 meter	363	48	29	142	173	58	206	45
$\leq$ 5 meter	411	0	169	2	226	5	244	7
$\leq$ 10 meter	411	0	171	0	229	2	250	1
$\leq$ 15 meter	411	0	171	0	231	0	251	0
Avg. Error (meter)	0.66		1.58		0.96		0.84	



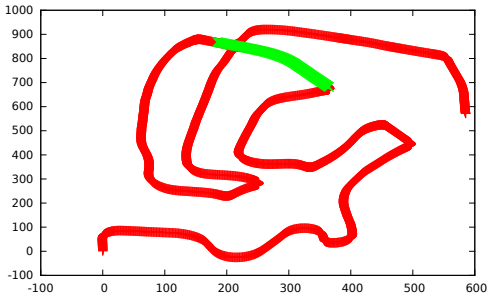
(a) Point feature matching.



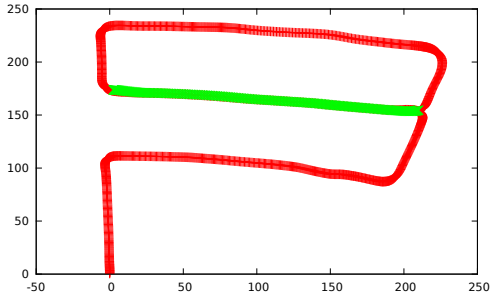
(b) Line feature matching.



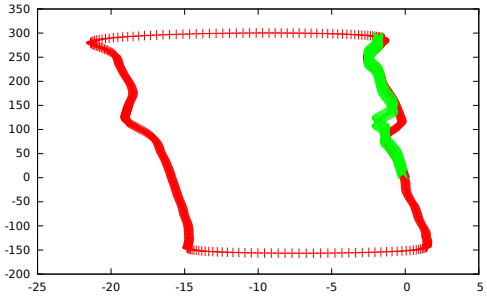
(c) KITTI00 dataset.



(d) KITTI02 dataset.



(e) KITTI05 dataset.



(f) KITTI06 dataset.

Figure 4.8: Point and line feature matching examples and loops detected by the proposed method in the KITTI dataset.

4.5 Conclusion

In this chapter, a modified vocabulary tree that combines multiple feature types for place recognition has been proposed. The ability of combining different feature types makes it possible for users to customize feature combination for various environments. Point feature and LBD line feature have been combined for loop closure

detection on ceiling images. The comparison results show that the proposed modified vocabulary tree that combines both point and line features outperforms the conventional vocabulary tree using each feature alone. To evaluate the performance of the proposed approach in outdoor environment, we have test our system on the KITTI dataset. The results have demonstrated the effectiveness of the proposed approach.

Chapter 5

Integrated Metric-topological Localization by Fusing Visual Odometry, Digital Map, and Place Recognition

5.1 Introduction

The work conducted in **Chapter 3** is originated from our previous study in [109] (a collaborative work with colleagues), where a localization framework that uses an available digital map and the on-road assumption is proposed to reduce the estimation error of MVO or SVO. The framework is designed based on Monte Carlo Localization, where visual odometry is used to generate particles (i.e. possible trajectories) with specific probabilities and shape matching between these trajectories and the digital map is applied to further weight the particles. Specifically, by assuming that the scale of monocular visual odometry follows uniform distribution in the interval $[a, b]$ and the raw translation follows Gaussian assumption, we

model the measurement of monocular visual odometry as a group of particles, which obey Uniform-Gaussian Distribution that covers measurement uncertainties including scale ambiguity and measurement randomness. The saliency of each particle can be obtained from the distribution to indicate raw measurement certainty of monocular visual odometry. Geometric map-assisted shape matching is implemented as the measurement model to assign consistency to the particles generated from the distribution. Both saliency and consistency are considered in particle weights determination. Furthermore, based on the statistical properties of the probability distribution, a parameter estimation scheme is proposed to narrow down the scale ambiguity of monocular visual odometry while resampling particles. Although good localization results have been obtained from the presented approach on KITTI dataset, there are still some challenges as follows:

1. The map assisted approach depends on the reliability of the digital map. However, most of the current digital maps have a update cycle. Positioning consistency can not be ensured in situations where a dated map (e.g. there is a newly-built road) is used.

2. The map-assisted approach only works in on-road scenarios due to the on-road assumption. It is desired to make the approach applicable to both on-road and off-road situations since the vehicle does not always run on the road.

3. The vehicle's initial position and orientation are unknown and the initialization process relies extremely on shape matching performance. A very large number of initial particles needs to be generated to cover all possible trajectories (with different starting positions, orientations and scales). Thus, the initialization process may be time consuming. Occasionally, the initialization result converges to wrong locations due to similar road shapes.

4. After initial position estimation, it is more reasonable to model the scale distribution as Gaussian instead of uniform, as the optimal estimation of true scale must be a particular point instead of an interval.

Although the metric position of the query image can not be computed, a rough topological location can be obtained through place recognition operation no matter the vehicle is on road or off road. Noticed that a rough position information helps significantly to the initialization process of [109], it is promising to incorporate place recognition into our geometric map-assisted approach.

In this chapter, we aim to localize a mobile vehicle equipped with one panoramic camera, one mono-camera and a digital map. Compared with [109], the main contributions of the integrated approach are:

1. A sensor fusion strategy is proposed to combine metric data from digital map-assisted VO with topological data from place recognition results. Within the strategy, a mutual check thread is implemented to measure the positioning consistency of different data sources and to determine whether topological results or metric results should be trusted.

2. A robust place recognition aided initialization scheme is presented to initialize the localization framework and the initialization time consumption is significantly reduced.

3. Gaussian probability assumption instead of uniform assumption is used to represent scale distribution of both SVO and MVO. The drift and scale ambiguity are modelled by a Gaussian-Gaussian distribution more robustly.

4. An on-road/off-road judging scheme is proposed such that the integrated approach is applicable for both on-road and off-road scenarios.

The remaining part of the chapter proceeds as below. Section 5.2 describes the integrated framework, where the details of all the tricks are emphasized. In Section 5.3, experiments on our own dataset are implemented and the results are discussed. Section 5.4 concludes the chapter.

5.2 Methodologies

In this section, topological and metric localization methods are explained separately first. Then an integrated framework is proposed based on their pros and cons.

5.2.1 Topological Localization Based on Place Recognition

As one typical topological localization approach, place recognition has been discussed in **Chapter 4**. Due to the issue of chronological order, we design a new place recognition framework in this section.

5.2.1.1 Database Creation

Lategahn et al. [110] introduced one visual feature, which they dub DIRD (Dird is an Illumination Robust Descriptor), among several millions that is best suited to represent places under illumination variations. Comparative experiments between DIRD and other descriptors demonstrated the effectiveness and efficiency of DIRD in place recognition. In this work, some changes are necessary to fit the particular application, and an improved DIRD is used to describe our panoramic images.

A vehicle equipped with one panoramic camera and a differential GPS (DGPS) travels the route to be recognized one or more times. As the vehicle travels the route, a database graph is created using the vehicle position at fixed distance intervals. Each node of the graph is annotated with the vehicle position and visual features. Vehicle positions are obtained from DGPS, while visual features are extracted from panoramic image. Both the vehicle position and visual features are stored in the database.

Fig. 5.1 shows the structure of the created database, which consists of the set $\mathcal{D} = \{f_k\}, k = 1, \cdots, K$, with components $f_k = \{\text{DIRD}_k, l_k\}$, where DIRD_k is the visual

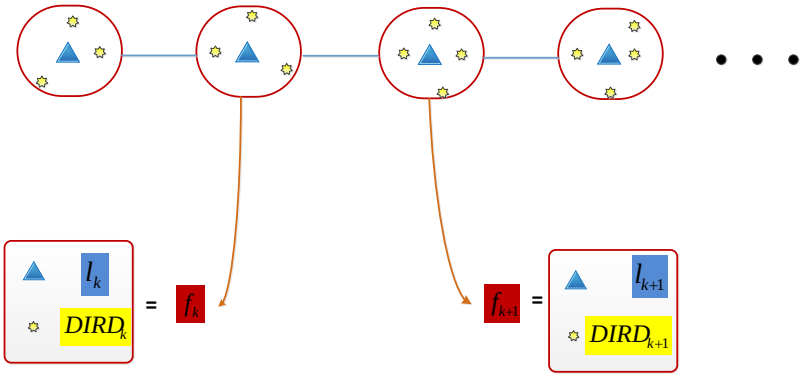


Figure 5.1: The database structure is shown. Panoramic appearance as well as vehicle position are stored in each node of the graph.

descriptor of the kth node; l_k is the ground truth location of the vehicle in the map.

5.2.1.2 Online Location Estimation

At run time, as the vehicle drives over the mapped routes, a search process is proposed to match the current panoramic view $\mathbf{I_t}$ against the pre-stored database $\mathcal{D}$ through feature matching. A column vector $\mathbf{d}_t$ of L1 distances can be computed by

$$\mathbf{d}_t = (\|\text{DIRD}_1 - \text{DIRD}_t\|_1, ..., \|\text{DIRD}_K - \text{DIRD}_t\|_1)^T. \tag{5.1}$$

Intuitively, the minimum argument of Eqn. (5.1) can be considered as the result of the place recognizer. However, matching the current query image with all the images stored in the database is quite time consuming. Furthermore, the place with the smallest L1 distance is not necessarily the best match due to dynamic objects, lateral shift or visual aliasing. In our work, a search window is used to restrict the matching range once place recognition has been initialized. Suppose f_{k_0} is the matching result at previous time step; w is the window size. This leads to a finite set $\mathcal{W}$ forming a sliding window around f_{k_0} placed in its center as:

$$\mathcal{W} = \{f_{k_0-\frac{w}{2}}, ..., f_{k_0}, ..., f_{k_0+\frac{w}{2}}\}. \tag{5.2}$$

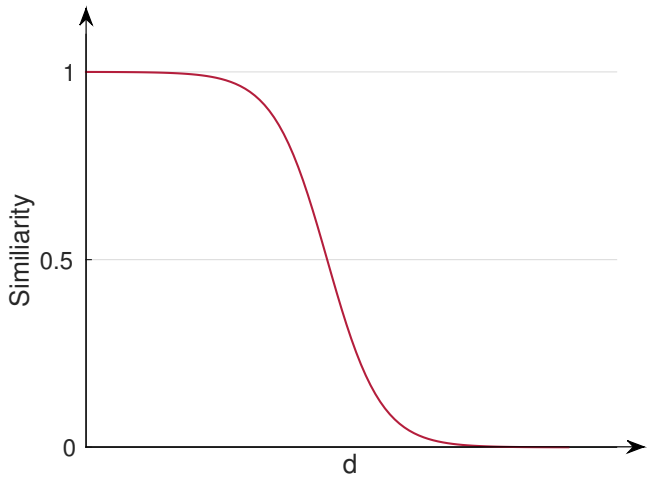


Figure 5.2: Graph of the logistic function used to translate all distance values into matching scores.

Then feature matching is only implemented inside this window and $\mathbf{d}_t$ becomes a vector of size w. The time consumption of this step is almost the same no matter how massive the database is. More importantly, the sequential consistency is maintained and positioning jump problem caused by perceptual aliasing is solved effectively. One thing should be noticed that global matching is still executed at intervals to correct failures of sliding window.

Besides, a parametrized logistic function logit$(\cdot)$ is used to convert all distances of $\mathbf{d_t}$ into matching score $\mathbf{p_t}$ in the range (0, 1). The logistic function is represented as

$$\text{logit}(d) = 1 - \frac{1}{1 + \exp\left(-\alpha(d - d_0)\right)}, \tag{5.3}$$

where d is one element of $\mathbf{d_t}$; d_0 is the L1 distance when the matching score is 0.5 and α denotes the steepness of this sigmoid curve. Fig. 5.2 shows the graph of this function. Through this sigmoid function, large distance values are translated to small matching scores (near zero) and small distance values are translated to large matching scores (near one).

Noted that our panoramic camera has six small ones. Each camera has a constant field of view. It often happens that part of these cameras are filled with moving

vehicles and pedestrians, which will cause partial appearance variations. If the panorama is considered as a whole, these partial appearance variations will severely disrupted feature matching performance. Instead, the images from each camera are considered separately in this work, so that partial appearance variations or aliasing will not determine the overall result. After computing the six matching score vectors $\{\mathbf{p}_t{}^1, ..., \mathbf{p}_t{}^6\}$ from Eqn. (5.3), their summation

$$\mathbf{p}_t = \sum_{n=1}^{6} \mathbf{p}_t{}^n \tag{5.4}$$

is considered as the overall similarities. Then the node who has the highest matching score is considered as the best matching result and the node's location

$$\mathrm{l}_k = \underset{f_k \in \mathcal{W}}{\operatorname{argmax}} \mathbf{p}_t \tag{5.5}$$

is treated as the position estimation of place recognizer.

Algorithm 2 gives pseudo-code for online location estimation.

Algorithm 2 Place Recognition

Input: Current panoramic $\mathbf{I_t}$
Output: Estimated location l_k

if initialized **then**
  if do global search **then**
    $\mathcal{W} \leftarrow \mathcal{D}$ /**search the whole database* $\mathcal{D}$*/
    $\mathbf{p}_t{}^n \leftarrow$ Eqn. (5.3) /**translate distance into matching probability* $\mathbf{p}_t{}^n, n \in 1, ..., 6$*/
    $\mathbf{p}_t \leftarrow$ Eqn. (5.4) /**generate overall similiarity* $\mathbf{p}_t$*/
    $l_k \leftarrow$ Eqn. (5.5) /**find the positioning result* l_k*/
  else if do window search **then**
    $\mathcal{W} \leftarrow$ Eqn. (5.2) /**search the local window**/
    $\mathbf{p}_t{}^n \leftarrow$ Eqn. (5.3) /**translate distance into matching probability* $\mathbf{p}_t{}^n, n \in 1, ..., 6$*/
    $\mathbf{p}_t \leftarrow$ Eqn. (5.4) /**generate overall similiarity* $\mathbf{p}_t$*/
    $l_k \leftarrow$ Eqn. (5.5) /**find the positioning result* l_k*/
  end if
else if not initialized **then**
  Initialization
end if

5.2.2 Metric Localization with Road-Constrained Visual Odometry Based on Gaussian-Gaussian Distribution

Compared with place recognition, the positioning result of visual odometry is a more accurate metric localization. Road-constrained VO is implemented to provide continuous pose estimation in [109]. In this work, most of the metric localization procedure follows our previous work [109] except that 1) Gaussian-Gaussian Distribution is used to generate possible MVO measurements; 2) Place recognition is implemented to assist initialization process.

5.2.2.1 Gaussian-Gaussian Distribution for Odometry Measurement Representation

Consider MVO measurement equation

$$\mathbf{t}_{k,k-1} = s_k \mathbf{t}^{\text{raw}}_{k,k-1} \tag{5.6}$$

where $\mathbf{t}_{k,k-1}$ and $\mathbf{t}^{\text{raw}}_{k,k-1}$ denote scaled translational vector and raw translational vector, respectively; $s_k \in \mathbb{R}^+$ is a scaling factor at time instant k.

As discussed earlier, in a monocular odometry localization system the drift and scale factor should be modelled properly. In Monte-Carlo localization, improper model of MVO measurement may generate low-quality particles such that localization performance is degraded. Conventional methods usually regard $\mathbf{t}_{k,k-1}$ as Gaussian. In our previous work [109], we model $\mathbf{t}_{k,k-1}$ based on product distribution, where s_k and $\mathbf{t}^{\text{raw}}_{k,k-1}$ are uniform-distributed and Gaussian-distributed random variables. Without a priori knowledge, it is reasonable to estimate scale s_k using uniform distribution. However, after obtaining the initial scale estimation, Gaussian-distributed s_k is preferred, as the true value of scale should be a point instead of an interval. With Gaussian assumption, the scaled translation vector from monocular visual odometry

is a product of two Gaussians. We know that a random variable from the product of two independent Gaussian random variables is not Gaussian except in some degenerate cases such as one random variable in the product being constant. Moreover, the distribution can't be modelled with a commonly used parametric distribution. In this chapter, for the sake of easy expression, we define a Gaussian-Gaussian distribution to model MVO measurement as follows:

Definition 5.1 (Gaussian-Gaussian Distribution, GGD)**.** *Given random variable S and random vector $\mathbf{X}$, the variate $\mathbf{Y} = S\mathbf{X}$ obeys Gaussian-Gaussian distribution $GG(ES, DS, E\mathbf{X}, \Sigma_X)$ if $S \sim N(ES, DS)$ and $\mathbf{X} \sim N(E\mathbf{X}, \Sigma_X)$, where $N(\cdot)$ denotes Gaussian distribution with corresponding expectation and covariance matrix (or variance).*

The expectation $E\mathbf{Y}$ and variance $D\mathbf{Y}$ of Gaussian-Gaussian distributed $\mathbf{Y}$ can be derived as

$$E\mathbf{Y} = ESE\mathbf{X} \tag{5.7}$$

$$DY_j = DX_j ES^2 + DSEX_j^2 + DSDX_j \tag{5.8}$$

where DS, DX_j and DY_j denote the variance of S, the variance of jth element in $\mathbf{X}$ and the variance of jth element in $\mathbf{Y}$, respectively.

Please note that it's for the sake of easy expression as well as intuitive representation for monocular odometry measurement, we define this Gaussian-Gaussian distribution. This distribution is employed to represent monocular odometry measurement uncertainties originated from drift issue and scale ambiguity. If we put the cloud concept aside, it has no fundamentally difference with Gaussian-Gaussian cloud proposed in **Chapter 3**. To emphasize the importance of the integration of topological and metric positioning, the concept of cloud is not exploited in this chapter. Particles which obey Gaussian-Gaussian distribution in this chapter are essentially the same with cloud drops that obey Gaussian-Gaussian cloud. By novelty, we mean the improvement compared with [109].

Based on the above definition, the scaled MVO measurement $t_{k,k-1}$ can be represented with Gaussian-Gaussian Distribution $GG(ES, DS, E\mathbf{X}, \Sigma_X)$. Given the four parameters of GGD, samples denoting possible MVO measurements can be generated. With GGD, both scale ambiguity and measurement randomness are modelled simultaneously.

Similar to our previous work, a Monte-Carlo framework is leveraged to combine MVO measurement and geometric map. Each particle is considered as one possible trajectory of the vehicle. After map preparation and initialization, particles are generated from VO raw measurement. By comparing the trajectory of each particle with road shape obtained from geometric map through chamfer matching, weights are assigned to each trajectory. Resampling is implemented to retain trajectories with higher weights for position and scale estimation. The estimated scale will be used to generate particles in the next time step. Through the processes of this framework, visual odometry drift is corrected and scale ambiguity is eliminated.

5.2.2.2 Place Recognition Aided Initialization

Pure odometry based localization system could not possibly give a global position estimation without an accurate initial global position and orientation. An initialization scheme is proposed in [109] , where a large number of initial particles is generated to cover all possible trajectories (with different starting positions, orientations and scales). And for each possible trajectory, one query edge map is created. Then an exhaustive shape matching between the road edge map and all the query edge maps is implemented to find the most possible localization hypotheses. Although experiments on KITTI benchmark had shown the effectiveness of shape matching based initialization, several challenges are still yet to be solved. First of all, shape matching performance depends on vehicle's trajectory and road conditions. Generally, the more complicate vehicle's trajectory is, the better performance will be. However, the vehicle's trajectory does not guarantee to meet such complexity. Pure

road shape assisted initialization cannot ensure position convergence to true value. Moreover, the number of initial particles relates to the map size. The larger the map size is, the more initial particles are needed. Thus the initialization process becomes time consuming when the searching region is very large.

In this work, the above challenges are solved by incorporating place recognition with road constrained approach. On the one hand, place recognition is firstly activated and the rough position estimation from place recognizer is then used to narrow down the initial searching region of the metric localization method. Only the nearby on-road area is considered as the possible starting position. On the other hand, the road direction can be obtained from the tangent orientation of the corresponding pixel point when generating road edge map. Assuming that initially the vehicle is parallel to the road, then the vehicle's starting orientation is always in accordance with the road direction. Hence, the number of the possible starting orientations is reduced dramatically. Then initial particles are generated to cover all the possible starting states. These particles will be fed into the shape matching scheme, from which the weight of each particle is obtained. A metric pose estimation is then computed after particle re-sampling. With this place recognition aided initialization, the large number of initial particles as well as the high requirement on road shape complexity in [109] is no longer needed. In other words, the initialization process becomes much easier. Algorithm 3 gives pseudo-code for our place recognition aided initialization.

5.2.3 Integrated Localization Strategy

In previous sections, both topological and metric methods are described to localize a mobile vehicle. In order to give full play to their advantages, an integrated strategy is presented in this section.

Fig. 5.3 demonstrates the flowchart of our integrated method, where no GPS or

Algorithm 3 Place Recognition Aided Initialization

Input: Place recognition estimation l_k, a set of historical MVO, number of initial particles N.

Output: Starting pose P_0 and initial scale s_0

1: **begin**
2: **for** $i \leftarrow 1$ to N **do**
3: *particle* $i \leftarrow$ possible trajectory /**starting position is normally distributed with mean* l_k*; starting orientation* $= (-1)^i *$ *road direction; initial scale is normally distributed with user defined mean* $\hat{s_0}$ */
4: $\mathbf{w}_i \leftarrow$ shape matcing /**compute the weight of each particle**/
5: **end for**
6: Resampling /**select the most probable starting pose and scale as* P_0 *and* s_0*/
7: **end**

other sensors is involved, and only visual information and an OpenStreetMap are utilized to accomplish the localization goal. First of all, a state variable indicating the state of the whole framework is introduced. At each time step, this state is firstly checked. If it is "NOT INITIALIZED", place recognition aided initialization scheme explained in previous section will be performed. Once initialization is succeeded, the state variable will be assigned as "OK" .

Another important feature in this framework is the on/off-road judging scheme, which makes the integrated approach applicable for both on-road and off-road scenarios. When generating the panoramic database, on-road frames are labelled as one and off-road frames are labelled as zero. At run time, given one panoramic frame, the label of the best matched database frame is considered as the current vehicle's state. If "ON ROAD" flag is false, only visual odometry is implemented since our road-constrained approach only works in on-road scenario. Otherwise, full road-constrained approach is implemented. Afterwards, a mutual check thread is implemented to determine if the estimation difference between place recognition and road-constrained threads is smaller than a user defined threshold ϵ. If yes, the metric estimation will be considered as the final pose estimation. Otherwise, the initialization step will be re-executed.

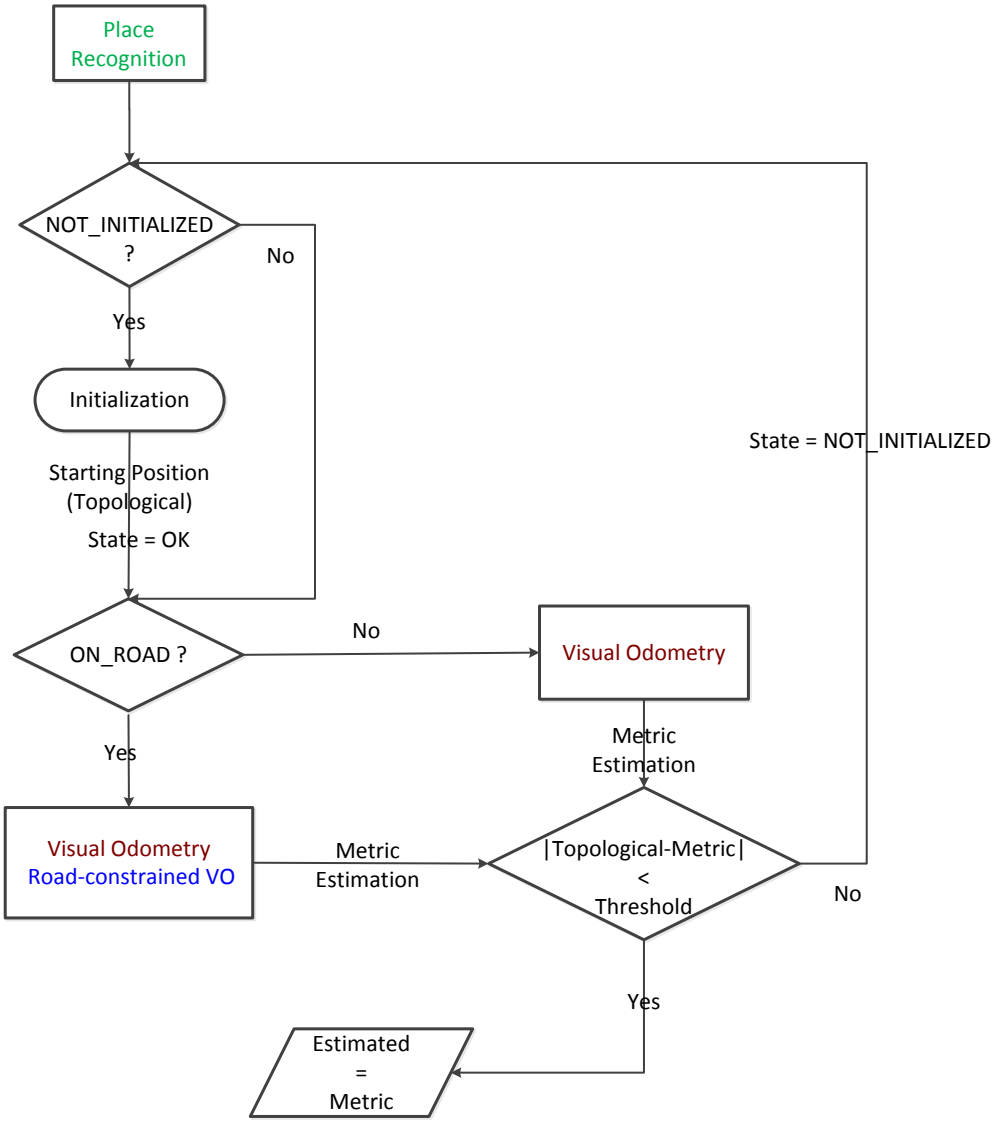


Figure 5.3: The flowchart of the integrated vehicle localization method.

5.3 Experimental Validation

Experimental validations conducted on real world datasets are explained in this section. We first illustrate the performance difference between Gaussian-Gaussian Distribution model and Uniform-Gaussian Distribution (UGD) model. Then the quantitative and qualitative positioning performance of the proposed framework are introduced. Experiments on initialization comparison are given in the last section.

5.3.1 Comparison Between GGD and UGD

Gaussian probability assumption instead of uniform assumption is used to represent scale distribution of MVO. This is the only improvement from GGD to UGD. In order to show their performance difference, evaluation experiments are conducted on KITTI benchmark. Scale estimation performance rather than the positioning result is leveraged to verify the advantage of the proposed localization frame. Some critical scale estimation results are shown in Fig. 5.4. The purple and red curves in the left are the upper and lower limits of the estimated scale interval from UGD. As can be seen, not all the estimated intervals cover the true scale. Since the scale estimation result is used for generating particles iteratively, once the scale interval $[a, b]$ estimated from the parameter estimation scheme does not contain the real one, the particle filter may diverge. The location estimation might be accurate but the scale ambiguity increases when the true scale is out of the estimated interval. This is an inherent flaw of UGD. To the contrary, GGD does not have this issue. The curves in the right show the estimated scale's expectation and variance from GGD. As can be seen, the expectation curves fit the ground truth well. The variance decreases rapidly at the beginning and becomes stable after a few iterations. Although, their are cases when the estimated means are a little far away from the true scales, particles generated in the next time step can still cover the truth with large probability as long as the true scales lie within three standard deviations of the mean. Thus, convergence can be ensured.

5.3.2 Localization Results

In order to evaluate the localization performance of the integrated strategy, experiments are conducted on our self-collected dataset.

Fig. 5.5a shows our evaluation mobile vehicle–Venus. It is a self designed four wheeled mobile robot. It can be navigated by joystick at human walking speed.

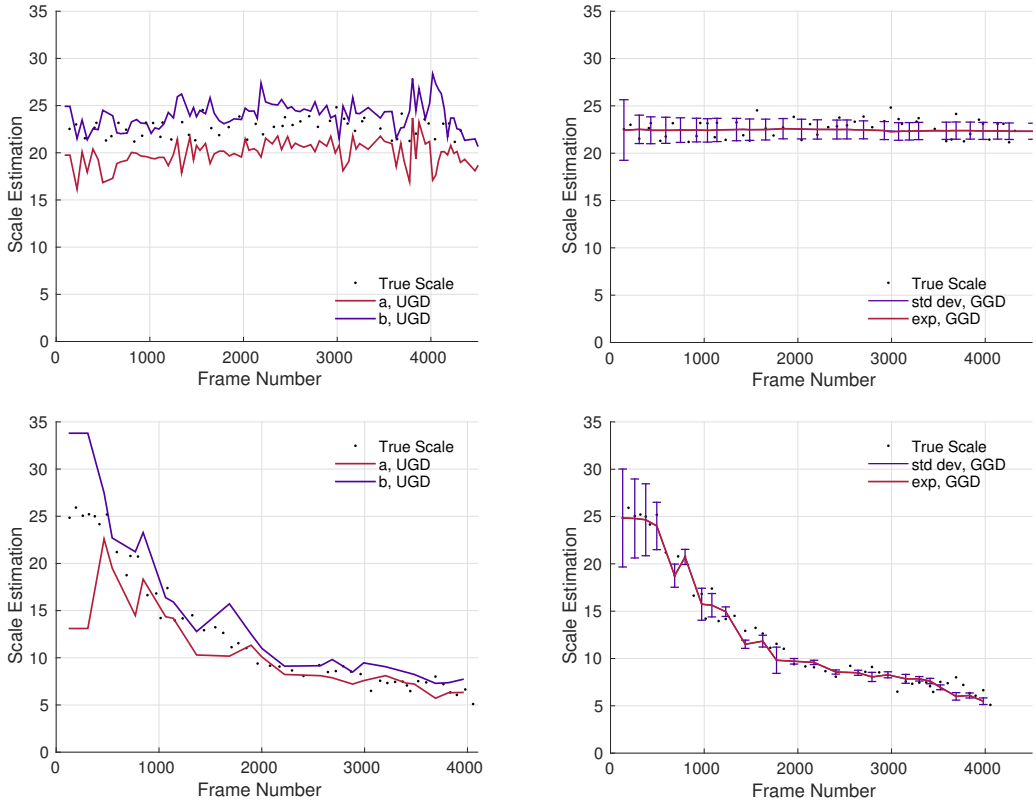
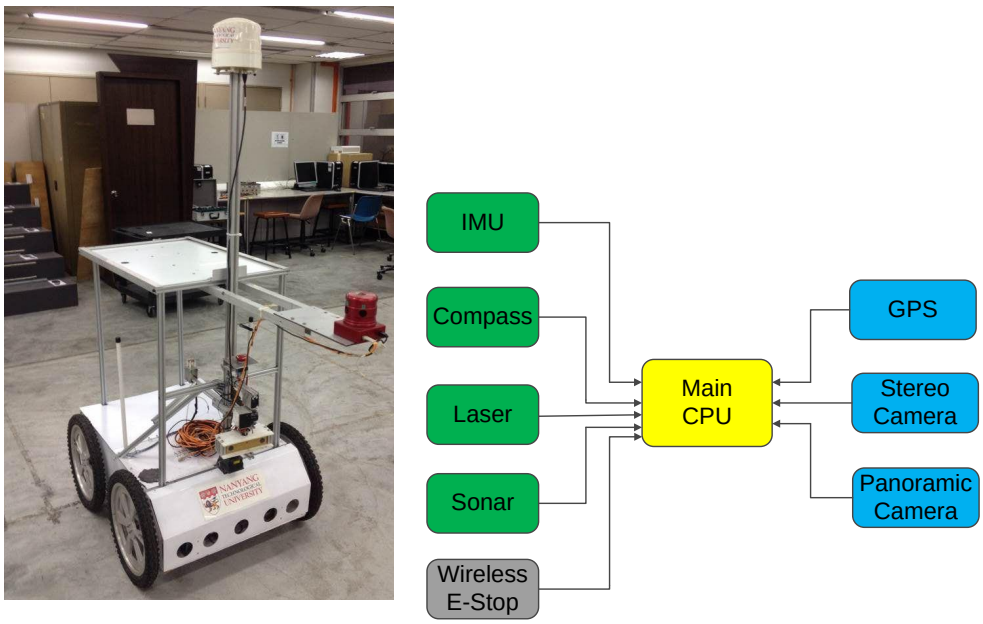


Figure 5.4: Scale estimation comparison between UGD and GGD on KITTI sequence 00 (up) and 08 (down). The left figures show the estimated scale's lower bound a and upper bound b of UGD. The right figures show the estimated scale's mean E_s and standard deviation σ_s of GGD. The ground truth scales are represented with black dots.

The robot is equipped with a DGPS to provide us with sub-meter level position ground truth. A stereo camera set is mounted on the robot and oriented forward. It is configured to acquire stereo frames at 10 Hz with a resolution of 1280 x 1024. The baseline of this stereo camera is configured at 30 cm to have a good effective depth range. The robot is also equipped with one Ladybug2 camera, which is used to capture panoramic view images. Other sensors like IMU, laser range finder and compass are also available from this platform. All the sensors are configured by one CPU. Fig. 5.5b shows the hardware block diagram.

Our dataset was collected by driving Venus around the campus of Nanyang Technological University. The testing routes have two parts with a 3000 meter length,

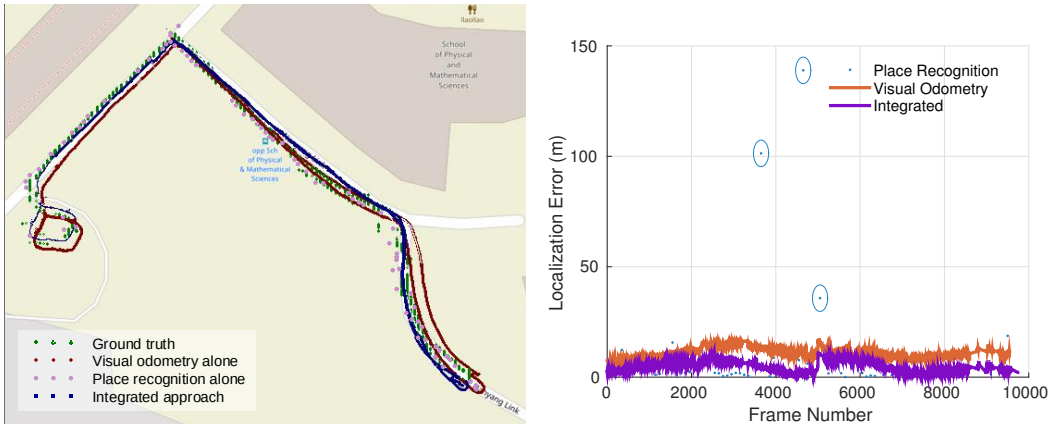


(a) Our mobile robot Venus. (b) Hardware block diagram.

Figure 5.5: Our platform and its hardware set-up.

including both on-road and off-road area. Fig. 5.6a is one screen shot of the positioning results on the first route. Trajectories estimated from DGPS, visual odometry, place recognition and the integrated approach are represented with different colors. As can be seen, visual odometry works fine in the first place, but it becomes worse and worse as the vehicle moves, especially when it comes to the off-road area. At the mean time, the positioning results from our integrated strategy are restricted to the road when the vehicle is travelling on the road. When the vehicle travels off the road, visual odometry takes over. A consistent good performance in both on-road and off-road areas is given from the proposed integrated strategy. Noted that the positioning results from road-constrained approach are not plotted because they are the same with the results from the integrated approach when the vehicle travels on road.

Localization error curves from each of the above methods are demonstrated in Fig. 5.6b. Place recognition updates at a low frequency while the other methods work continuously. The three circles are situations where re-initializations are



(a) Qualitative comparison. (b) Localization error.

Figure 5.6: Localization performance comparison.

Table 5.1: Quantitative results of place recognition, visual odometry, road-constrained approach and our integrated strategy, respectively.

	Place Recognition	Visual Odometry	Road-Constrained	Integrated
Avg Error (m)	9.0	10.9	2.9 (only on road)	2.9
Median Error (m)	2.7	10.7	4.6 (only on road)	4.3
Max Error (m)	138.9	19.1	11.1 (only on road)	11.1

activated due to the huge positioning difference between place recognition and road-constrained thread. Noted that place recognition's bad performance plays a leading role in the three re-initializations. The database and query images for the three circles are shown in Fig. 5.7. As can be seen, great appearance variations have occurred. The query views for the first two circles are blocked by school buses and the grass has withered in the third case. All these disturbances cause the huge place recognition errors.

Table 5.1 lists the quantitative results of place recognition, visual odometry, road-constrained method and the proposed integrated approach, respectively. The whole position error of the integrated is less than 3 meter over the 3 km run, while the other methods either has a much larger error or has restrictions to use. Considering that all the roads in geometric map are modelled with centre lines, positioning error in the lateral direction of the road could not be eliminated. A 3 meter positioning error is quite acceptable.

Figure 5.7: The database and query images of the three circles marked in Fig. 5.6b. The top images show the views obtained from the database for the estimated vehicle location. The bottom images show the views of the query image of the current position. Only one view of the six cameras is shown here.

5.3.3 Initialization Analysis

The initialization process is in charge of finding the initial state (starting position, starting orientation and initial scale). In the particle framework, a lot of initial particles need to be generated to cover all the possible initial states. This makes initialization process be the most time consuming part. In [109], the number of initial particles is largely determined by the map size. The red curve of Fig. 5.8 shows the relationship between the initialization time of [109] and the map size. It can be seen that as the map size increases, the initialization time grows in quadratic function. It is easy to understand this as the number of possible starting positions increases in a square number when the map size increases. At the same time, the number of the initial states of the proposed place recognition aided initialization no longer

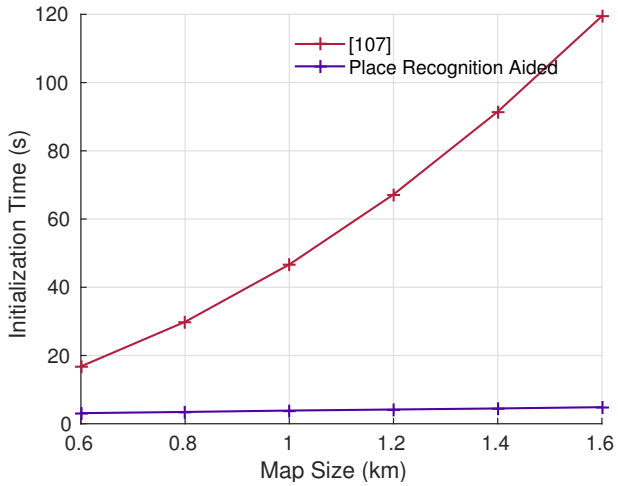


Figure 5.8: Initialization time comparison between [109] and the place recognition aided.

depends on the map size. The purple curve of Fig. 5.8 shows the corresponding time variation. As can be seen, the proposed has a slightly increased initialization time due to the growing place recognition database. But the time consumption (around 3 seconds) is far less than [109]. The integrated approach is implemented in C++. And all the experiments run on a mobile workstation with an i7-4710MQ processor.

As discussed in Section 5.2.2.2, the vehicle has to travel a minimum distance before the vehicle can be localized. Table 5.2 shows the minimum trajectory length and particle numbers to guarantee the success of initialization on KITTI benchmark. Here, maps with size of 1 km are used for all the sequences. "P" indicates our place recognition aided initialization method. "X" indicates sequences which failed on initialization. Since no panoramic image is available from KITTI odometry dataset, a pseudo place recognition is performed like this: all the right images in KITTI dataset are considered as visual database and the left images are matched with visual database at run-time. As can be seen, no matter for [109] or place recognition aided, different sequences need different minimum trajectory lengths due to different road conditions. The initial trajectories of sequence 03, 05 and 07 are either straight or less distinctive. Thus slightly larger values are needed. However, in general, a much smaller historical trajectory is required for the place recognition aided. Besides,

Table 5.2: Minimum trajectory length and particle numbers to guarantee the success of initialization on KITTI benchmark. "P" indicates place recognition aided initialization. "X" indicates sequences which failed on initialization.

Sequence	[109]		P	
	Min length (m)	Min particles	Min length (m)	Min particles
KITTI 00	350	50 k	200	3 k
KITTI 01	X	X	X	X
KITTI 02	350	50 k	200	3 k
KITTI 03	400	50 k	200	3 k
KITTI 04	X	X	X	X
KITTI 05	400	50 k	200	3 k
KITTI 06	X	X	300	3 k
KITTI 07	400	50 k	200	3 k
KITTI 08	350	50 k	200	3 k
KITTI 09	350	50 k	200	3 k
KITTI 10	X	X	200	3 k

sequence 06 and 10, which did not localize in [109] are also successfully initialized. The minimum number of particles needed are dropped significantly, from 50 k down to 3 k. All of these prove that the place recognition aided initialization approach is much more robust and efficient.

5.4 Conclusions

In this chapter, an integrated strategy has been proposed to localize a mobile vehicle equipped with one panoramic camera, one mono-camera and one digital map. Place recognition, visual odometry and road-constrained approaches have been incorporated into one framework. With in this framework, place recognition plays a role of topological localization and assists initialization process. Road-constrained is responsible for on-road localization while visual odometry handles the off-road scenario. Gaussian assumption instead of uniform assumption has been proposed to model the scale distribution of monocular visual odometry. Evaluation results show that the proposed framework is much more accurate than separate implementation of these methods.

Chapter 6

GPS, Odometry, and Map Fusion for Vehicle Positioning Using Potential Function

6.1 Introduction

In previous chapters, we focus on the development of vision-based localization without the participation of other sensors. For vehicles under complex environments, single information source is far from enough in positioning accuracy and robustness. For example, as an absolute positioning sensor, Global Positioning System (GPS) performs well in large-scale localization but suffers from temporary signal lost and small-scale inaccuracy[111], [112]. Odometry and inertial sensors are based on dead reckoning or relative positioning, whose accumulative error makes estimated location drift over time[113], [114]. Vision-based approaches generally match the landmarks in current frame with database map [115], [116], while the requirement of pre-established database limits its application. Digital maps, providing constraints of the vehicle on the road, are increasingly used as an additional information source

especially in urban areas[117], and map matching is usually a necessary procedure to align other measurements with roads. Data fusion approach is required to tackle vehicle localization problems when multiple information sources are available.

Most of the current research is on the lower levels of fusion [118], and several fusion methods are proposed for vehicle positioning. So far, Kalman filter and its varieties (e.g., extended Kalman filter, information filter, sigma point Kalman filter and multiple model algorithms) are still the most popular fusion methods [119], [120], [121], [122]. The Kalman filter related approaches need the information of system model and process noise, which are sometimes unavailable. When information conflicts with each other, it is hard to detect and provide reasonable positioning results immediately. Moreover, the Kalman filter related approaches suffer from the errors introduced in the linearisation process in real applications.

In this chapter, we propose a fusion approach by introducing the concept of potential functions to form potential wells and potential trenches. Under this fusion frame, the data fusion problem can be converted to a minimum searching problem, where each information source produces corresponding potential field according to data properties, and the spatial point with the lowest potential is estimated as fused position. The main contributions of this chapter are listed as follows:

- An insightful and straightforward data fusion frame based on potential function is proposed and implemented in vehicle positioning problem. A visual odometry, GPS, and two-dimensional digital road maps are integrated inherently without map matching algorithms.
- The proposed scheme does not rely on any kinematic model and can be easily generalized to other data fusion problems.
- The proposed approach is extensively and successfully verified on real world dataset.

This chapter proceeds as follows. In Section 6.2, the concepts of potential functions,

potential wells and potential trenches are introduced, followed by the elaboration of potential field creation according to information properties. Section 6.3 details the potential function based fusion approach for vehicle positioning. The experimental results are presented in Section 6.4. Finally, conclusions are made in Section 6.5.

6.2 Potential Wells and Potential Trenches

The concept of artificial potential field was first proposed in [123] to deal with obstacle avoidance for manipulators and mobile robots. The potential trenches were introduced in [124] in multi-robot formation, where the goals create attractive fields and obstacles create repulsive fields. The sum of attractive fields and repulsive fields make the robot move in configuration space. In this chapter, the sources of potential fields are data, and the fusion results are searched in data space instead. Before proceeding further, the following concepts are defined.

Definition 6.1 (Data space)**.** *An n-dimensional data space in the fusion frame $\mathcal{D}^n$ is the n-dimensional vector space of all possible data and fusion results under the frame.*

We only consider fusion problem on a single level [125]; thus data and fusion results should be compatible in one data space. Data and fusion results in a data space can be represented as geometric shapes, such as points, lines, surfaces or hypersurfaces. For example, in a two-dimensional data space, data measured by a position sensor can be represented as a point (x, y), and an on-the-road constraint can be expressed as a curve $c(x, y) = 0$.

Definition 6.2 (Potential function [126])**.** *A potential function in n-dimensional data space is a differentiable real-valued function $U : \mathbb{R}^n \to \mathbb{R}$. Given a vector field $\mathbf{F}$, the potential U is defined such that*

$$\mathbf{F} = -\nabla U. \tag{6.1}$$

Both potential function U and vector field $\mathbf{F}$ can be used to describe a data space. Considering the convenience of scalar operations, we mainly design U according to data properties. There are two types of potential functions defined in this chapter: potential wells and potential trenches.

Definition 6.3 (Potential well functions)**.** *Given a point* $\mathbf{q}^s = (q_1^s, q_2^s, ..., q_n^s)^T$ *in* $\mathcal{D}^n$*, the potential well function is defined as* $U_w(\mathbf{q}) = \alpha_w f_w(\mathbf{d}_w)$*, where* $\alpha_w > 0$ *is a user defined parameter,* $f_w(\cdot)$ *is the potential construction function and* $\mathbf{d}_w = \mathbf{q} - \mathbf{q}^s$ *is the distance vector between* $\mathbf{q}^s$ *and* $\mathbf{q}$*. The point* $\mathbf{q}^s$ *is called the source of the potential well.*

Definition 6.4 (Distance between a shape and a point)**.** *Given a shape* $c^s(\mathbf{q}) = 0 \in \mathcal{D}^n$ *and a point* $\mathbf{q} = (q_1, q_2, \cdots, q_n)^T$*, the distance between the shape* $c^s(\mathbf{q}) = 0$ *and the point* $\mathbf{q}$ *is defined as the radius of the hyper-sphere centered at* $\mathbf{q}$ *and tangent to* $c^s(\mathbf{q}) = 0$*.*

Definition 6.5 (Potential trench functions)**.** *Given a shape* $c^s(\mathbf{q}) = 0 \in \mathcal{D}^n$*, the potential trench function is defined as* $U_t(\mathbf{q}) = \alpha_t f_t(d_t)$*, where* $\alpha_t > 0$ *is a user defined parameter,* $f_t(\cdot)$ *is the potential construction function and* d_t *is the distance between the shape* $c^s(\mathbf{q}) = 0$ *and the point* $\mathbf{q}$*. The shape* $c^s(\mathbf{q}) = 0$ *is called the source of the potential trench.*

To visually illustrate potential well and potential trench, two examples are created in Fig. 6.1.

6.2.1 Potential Function Creation

Based on aforementioned definitions, in order to create a potential well $U_{\mathrm{w}}(\mathbf{q}) = \alpha_{\mathrm{w}} f_{\mathrm{w}}(\mathbf{d}_{\mathrm{w}})$ and a potential trench $U_{\mathrm{t}}(\mathbf{q}) = \alpha_{\mathrm{t}} f_{\mathrm{t}}(d_{\mathrm{t}})$ in data space, it is necessary to determine parameters α_{w}, α_{t}, sources $\mathbf{q}^{\mathrm{s}}$, $c^s(\mathbf{q}) = 0$, and potential construction functions $f_{\mathrm{w}}(\cdot)$, $f_{\mathrm{t}}(\cdot)$. The potential construction functions map the distance between a source and a point in data space to a real number. According to Definition

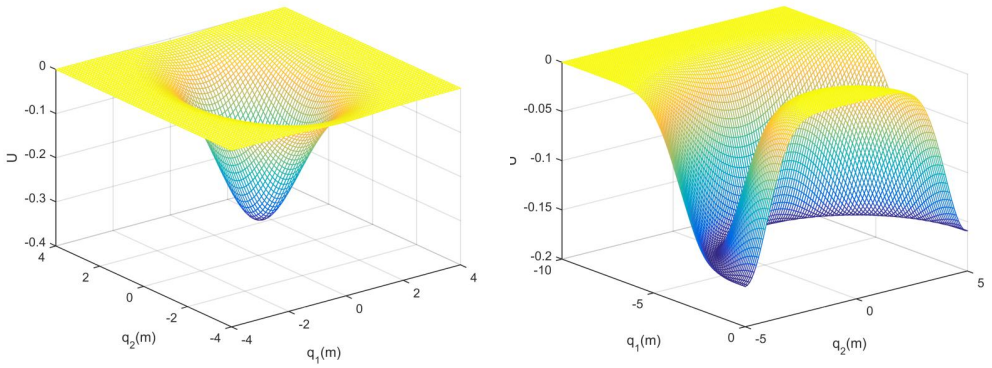


Figure 6.1: Examples of a potential well and a potential trench in 2D data space. (a) Potential well with $\alpha_{\mathrm{w}} = 1$, construction function $f(\mathbf{d}_{\mathrm{w}}) = -(2\pi)^{-1}|\mathbf{D}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\mathbf{d}_{\mathrm{w}}^{T}\mathbf{D}^{-1}\mathbf{d}_{\mathrm{w}}\right)$, source $\mathbf{q}^{\mathrm{s}} = (0,0)$ and $\mathbf{D} = \mathrm{diag}(1,2)$. (b) Potential trench with $\alpha_{\mathrm{w}} = 1$, construction function $f(d_{\mathrm{t}}) = -\sigma^{-1}(2\pi)^{-\frac{1}{2}}\exp\left(-\frac{d_{\mathrm{t}}^2}{2\sigma^2}\right)$, source $c^{\mathrm{s}}(q_1, q_2) = q_1^2 + q_2^2 - 25 = 0$ and $\sigma = 1$.

6.2, 6.3 and 6.5, given potential functions $f_{\mathrm{w}}(\cdot)$ and $f_{\mathrm{t}}(\cdot)$, $U_{\mathrm{w}}(\mathbf{q})$ and $U_{\mathrm{t}}(\mathbf{q})$ can be obtained self-consistently. Without loss of generality, the default potential field in data space is initialized to zero, and the point with minimum potential is regarded as the fusion result. Thus, $f_{\mathrm{w}}(\cdot) \leq 0$ and $f_{\mathrm{t}}(\cdot) \leq 0$ should be ensured, and the potentials with $d_{\mathrm{t}} = 0$ or $\mathbf{d}_{\mathrm{w}} = \mathbf{0}$ are required to be the minimum for corresponding potential functions. To summarize, any form of f can be created if f satisfies: 1) f is a potential function; 2) $f \leq 0$; 3) f reaches the minimum at 0.

Due to the distinct properties between potential wells and trenches, we will discuss how to create f with regard to measurements and constraints separately.

6.2.1.1 Potential Well as Measurement

A measurement can be represented as a point in data space, where the source of a potential well is also a point. It is natural to create potential wells representing measurements from different sensors. There are basically two ways to create potential wells: 1) creating multiple potential wells at a time instant, one for each information source; and 2) creating an individual potential well for all measurement

data. In this chapter, the second approach is preferred, as the first method leads to a mixture model which does not conform to reality, and an individual potential well contributes to local minimum avoidance. The following three assumptions are made before we proceed:

Assumption 4 (Synchronous fusion)**.** *The data from different information sources are synchronized.*

Assumption 5. *The data from information source i at the same time follow Gaussian distribution with mean $\mathbf{q}_i^s$ and covariance matrix $\mathbf{D}_i$.*

Assumption 6. *The fused data $\mathbf{y}$ can be represented as the linear combination of different independent data sources: $\mathbf{y} = \sum_{i=1}^{m} \mathbf{W}_i \mathbf{y}_i$, where $\mathbf{W}_i$ denotes weights satisfying $\sum_{i=1}^{m} \mathbf{W}_i = \mathbf{I}$, $\mathbf{y}_i$ $(i = 1, \cdots, m)$ are measurement data from m independent information sources, and $\mathbf{I}$ is the identity matrix.*

According to the assumptions, it follows that

$$\mathbf{y}_i \sim N(\mathbf{q}_i^{\mathrm{s}}, \mathbf{D}_i), \tag{6.2}$$

where $\mathbf{q}_i^{\mathrm{s}}$ is the unknown true value, and $\mathbf{D}_i$ is the covariance matrix for each information source. The matrix $\mathbf{D}_i$ can be estimated from sensor properties or observation data. It has been proved that

$$\mathbf{y} \sim N(\sum_{i=1}^{m} \mathbf{W}_i \mathbf{q}_i^{\mathrm{s}}, \sum_{i=1}^{m} \mathbf{W}_i \mathbf{D}_i \mathbf{W}_i^T) \sim N(\mathbf{q}^{\mathrm{s}}, \mathbf{D}). \tag{6.3}$$

As we already know $\mathbf{D}_i$, the problem turns to estimate $\mathbf{q}^{\mathrm{s}}$ and $\mathbf{W}_i$ in order to obtain fused data.

Proposition 6.1. *Given data $\mathbf{y}_i$ and covariance matrix $\mathbf{D}_i (i = 1, \cdots, m)$ from m independent information sources, the fused data $\mathbf{y}$ can be obtained from*

$$\mathbf{y} = \hat{\mathbf{q}}^s = \sum_{i=1}^{m} \mathbf{W}_i \mathbf{y}_i \tag{6.4}$$

with weights

$$\mathbf{W}_i = \left(\sum_{i=1}^{m} \mathbf{D}_i^{-1}\right)^{-1} \mathbf{D}_i^{-1} \tag{6.5}$$

Proof According to the assumptions, the fused data follow Gaussian distribution. By applying the maximum likelihood method, we may estimate $\mathbf{q}^{\mathrm{s}}$ from maximizing likelihood function

$$L(\mathbf{q}^{\mathrm{s}}|\mathbf{y}_1, \cdots, \mathbf{y}_m) = \prod_{i=1}^{m} p(\mathbf{y}_i|\mathbf{q}^{\mathrm{s}}, \mathbf{D}_i), \tag{6.6}$$

where
$p(\mathbf{y}_i|\mathbf{q}^{\mathrm{s}}, \mathbf{D}_i) = (2\pi)^{-\frac{n}{2}} |\mathbf{D}_i|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})^T \mathbf{D}_i^{-1}(\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})\right)$. By substituting $p(\mathbf{y}_i|\mathbf{q}^{\mathrm{s}}, \mathbf{D}_i)$ into (6.6), the likelihood function is written as

$$L(\mathbf{q}^{\mathrm{s}}|\mathbf{y}_1, \cdots, \mathbf{y}_m) = (2\pi)^{-\frac{mn}{2}} \prod_{i=1}^{m} |\mathbf{D}_i|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{m} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})^T \mathbf{D}_i^{-1} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})\right). \tag{6.7}$$

The log-likelihood function is

$$\ln(L) = \ln\left((2\pi)^{-\frac{mn}{2}}\right) + \sum_{i=1}^{m} \left(\ln\left(|\mathbf{D}_i|^{-\frac{1}{2}}\right) - \frac{1}{2}(\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})^T \mathbf{D}_i^{-1} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})\right).$$

Taking its derivative w.r.t. $\mathbf{q}^{\mathrm{s}}$ and setting it to zero we have

$$\frac{\mathrm{d}\ln(L)}{\mathrm{d}\mathbf{q}^{\mathrm{s}}} = -\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}\mathbf{q}^{\mathrm{s}}}\left(\sum_{i=1}^{m} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})^T \mathbf{D}_i^{-1} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}})\right) \tag{6.8}$$

$$= \sum_{i=1}^{m} \mathbf{D}_i^{-1} (\mathbf{y}_i - \mathbf{q}^{\mathrm{s}}) \tag{6.9}$$

$$= \mathbf{0} \tag{6.10}$$

$$\Rightarrow \hat{\mathbf{q}}^{\mathrm{s}} = \left(\sum_{i=1}^{m} \mathbf{D}_i^{-1}\right)^{-1} \sum_{i=1}^{m} \mathbf{D}_i^{-1} \mathbf{y}_i. \tag{6.11}$$

The result is similar to a weighted average calculation. From the result we may find that i) For a specific measurement with a small covariance, the weight will be large,

indicating that the measurement is more trusted. ii) For m information sources with the same covariance matrix, the weight will be equal. The fusion strategy degenerates to the simple average method.

Since we have obtained the fused measurement, we proceed to create a potential well based on fused data. We define the negative Gaussian potential well in $\mathcal{D}^n$ as

$$U_{\mathrm{w}}(\mathbf{q}) = \alpha_{\mathrm{w}} f_{\mathrm{w}}(\mathbf{d}_{\mathrm{w}}), \tag{6.12}$$

$$f_{\mathrm{w}}(\mathbf{d}_{\mathrm{w}}) = -\frac{1}{(2\pi)^{\frac{n}{2}} |\mathbf{D}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{d}_{\mathrm{w}}^T \mathbf{D}^{-1} \mathbf{d}_{\mathrm{w}}\right), \tag{6.13}$$

where $\mathbf{d}_{\mathrm{w}} = \mathbf{q}^{\mathrm{s}} - \mathbf{q}$. The covariance matrix $\mathbf{D}$ can be calculated from (6.3), and the source $\mathbf{q}^{\mathrm{s}}$ can be estimated according to (6.4) and (6.5).

6.2.1.2 Potential Trench as Constraints

The potential trench can be created to reflect constraints in data space. We consider a single constraint first: suppose we have the constraint $c^{\mathrm{s}}(\mathbf{q}) = 0$ in $\mathcal{D}^3$, referring to Definition 6.4, the distance between $c^{\mathrm{s}}(\mathbf{q}) = 0$ and $\mathbf{q}$ can be solved from the following general equation:

$$d_{\mathrm{t}} = \min \sqrt{(\mathbf{q}^{\mathrm{s}} - \mathbf{q})^T (\mathbf{q}^{\mathrm{s}} - \mathbf{q})} \quad \text{subject to } c^{\mathrm{s}}(\mathbf{q}^{\mathrm{s}}) = 0. \tag{6.14}$$

As the constraint is another kind of information source, which may not be absolutely trusted, we define the negative Gaussian potential trench in $\mathcal{D}^n$ as

$$U_{\mathrm{t}}(\mathbf{q}) = \alpha_{\mathrm{t}} f_{\mathrm{t}}(d_{\mathrm{t}}), \tag{6.15}$$

$$f_{\mathrm{t}}(d_{\mathrm{t}}) = -\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_{\mathrm{t}}^2}{2\sigma^2}\right), \tag{6.16}$$

where the confidence coefficient σ directly influences how much we trust the constraint.

Now let us consider multiple constraints. Similar to multiple data sources in potential well creation, there are two ways to deal with multiple constraints: 1) creating multiple potential trenches; and 2) creating a single trench for the current constraint, meanwhile switching the constraint based on specific conditions varying with time. In vehicle localization, the second approach is preferred as the vehicle is always constrained on one specific piece of road at a moment. Also, more potential functions may decrease real-time performance and increase the chance of generating local minimums. The switching strategy will be introduced in Section 6.3.3.

6.2.2 Minimum Searching

The potential field in data space may be written as the sum of all potential functions:

$$U(\mathbf{q}) = U_{\mathrm{w}}(\mathbf{q}) + U_{\mathrm{t}}(\mathbf{q}), \tag{6.17}$$

with user-defined parameters α_{w}, $\alpha_{\mathrm{t}} > 0$. The points with local minimum potentials in data space are regarded as the candidates of the fusion result.

The most straight-forward method to find local minimum potentials is by taking derivatives of $U(\mathbf{q})$ with respect to $\mathbf{q}$ and choosing the local minimums among critical points. Due to the difficulty of obtaining analytical solutions from transcendental equations $\frac{\mathrm{d}U(\mathbf{q})}{\mathrm{d}\mathbf{q}} = \mathbf{0}$, the quasi-Newton methods [127] are used in this work.

6.3 Potential-Function Based Fusion for Vehicle Positioning

The localization system consists of several data sources with different reference frames. These reference frames are defined here [120]: visual odometry uses a right-handed camera frame whose origin is at the center of the left camera, x_{c} axis points

to the center of the right camera and y_c axis points downward along the image plane. GPS and digital maps output data in WGS-84 coordinate system with longitude and latitude (altitude is only available for GPS). All the coordinates in camera frame and WGS-84 frame will be converted or projected to local Cartesian coordinates in the East-North-Up (ENU) working frame O_{xyz}.

6.3.1 Information Sources and Sensors

6.3.1.1 Visual Odometry

The visual odometry algorithm in [105] is implemented in this chapter. Due to the recursive nature of VO drift error [128], we model the covariance matrix $\mathbf{D}_{\mathrm{VO}}$ as

$$\mathbf{D}_{\mathrm{VO}} = \mathbf{D}_0 \eta^{t+\beta * r}, \tag{6.18}$$

where $\mathbf{D}_0$ is the initial covariance matrix, $\eta > 1$ is a user-defined parameter, t and r are the current displacement and angle displacement measured by visual odometry, β is a weighting parameter.

6.3.1.2 Digital Maps

Herein, digital maps providing road constraints are used to create potential trenches. Digital maps are generally represented as nodes and ways, where each node consists of its ID, latitude, longitude and each way is an ordered list of nodes. After projection, digital maps are flattened, with nodes and ways represented in two-dimensional Cartesian space without z axis. A general linear road constraint in a two-dimensional map is a plane in three-dimensional data space, and can be represented as $c^{\mathrm{s}}(\mathbf{q}) = p_{\mathrm{a}}x + p_{\mathrm{b}}y + p_{\mathrm{d}} = 0$ where $p_{\mathrm{a}},\ p_{\mathrm{b}},\ p_{\mathrm{c}}$ are shaping parameters, the potential generated by $c^{\mathrm{s}}(\mathbf{q}) = 0$ at point $\mathbf{q}(x, y, z)$ can be calculated from (6.15)

and (6.16) where

$$d_{\mathrm{t}} = \frac{|p_{\mathrm{a}}x + p_{\mathrm{b}}y + p_{\mathrm{d}}|}{\sqrt{p_{\mathrm{a}}^2 + p_{\mathrm{b}}^2}}. \tag{6.19}$$

Sometimes, the road cannot be simply modelled as linear segments. Compared to linear models, arc models keep positioning results smoother because of less modelling error and fewer switches between constraints. An arc model with center $(x_{\mathrm{c}}, y_{\mathrm{c}})$ and radius r is represented as $c^{\mathrm{s}}(\mathbf{q}) = (x - x_{\mathrm{c}})^2 + (y - y_{\mathrm{c}})^2 - r^2 = 0$, the potential at point $\mathbf{q}(x, y, z)$ can be calculated similarly from (6.15) and (6.16) where

$$d_{\mathrm{t}} = |\sqrt{(x - x_{\mathrm{c}})^2 + (y - y_{\mathrm{c}})^2} - r|. \tag{6.20}$$

The confidence coefficient σ in (6.16) is selected according to road condition and experience. In general, for a wider road, σ should be larger to indicate the less confidence of vehicle's existence on the road center line.

6.3.1.3 GPS

Similar to digital maps, GPS outputs vehicle's position with latitude, longitude and altitude. The coordinate transformation makes it compatible with local ENU coordinates. The covariance of GPS measurement $\mathbf{D}_{\mathrm{GPS}}$ can be obtained from National Marine Electronics Association (NMEA) messages GST, as it provides position error statistics such as standard deviations for latitude, longitude and height [120].

6.3.2 Potential Representation

With multiple information sources as above, based on (6.17), the total potential field in data space is represented with (6.21).

$$\begin{aligned} U(\mathbf{q}) =& \alpha_{\mathrm{w}} f_{\mathrm{w}}(\mathbf{d}_{\mathrm{w}}) + \alpha_{\mathrm{t}} f_{\mathrm{t}}(d_{\mathrm{t}}) \\ =& -\frac{\alpha_{\mathrm{w}}}{(2\pi)^{\frac{3}{2}} |\mathbf{D}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{d}_{\mathrm{w}}^T \mathbf{D}^{-1} \mathbf{d}_{\mathrm{w}}\right) - \frac{\alpha_{\mathrm{t}}}{\sigma\sqrt{2\pi}} \exp\left(-\frac{d_{\mathrm{t}}^2}{2\sigma^2}\right) \end{aligned} \tag{6.21}$$

$$=\begin{cases} -\frac{\alpha_{\mathrm{w}}}{(2\pi)^{\frac{3}{2}}|\mathbf{D}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\left(\mathbf{q}-\mathbf{q}^{\mathrm{s}}\right)^{T}\mathbf{D}^{-1}\left(\mathbf{q}-\mathbf{q}^{\mathrm{s}}\right)\right)-\frac{\alpha_{\mathrm{t}}}{\sigma\sqrt{2\pi}}\exp\left(\frac{\left(p_{\mathrm{a}}x+p_{\mathrm{b}}y+p_{\mathrm{d}}\right)^{2}}{-2\sigma^{2}\left(p_{\mathrm{a}}^{2}+p_{\mathrm{b}}^{2}\right)}\right) \\ -\frac{\alpha_{\mathrm{w}}}{(2\pi)^{\frac{3}{2}}|\mathbf{D}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\left(\mathbf{q}-\mathbf{q}^{\mathrm{s}}\right)^{T}\mathbf{D}^{-1}\left(\mathbf{q}-\mathbf{q}^{\mathrm{s}}\right)\right)-\frac{\alpha_{\mathrm{t}}}{\sigma\sqrt{2\pi}}\exp\left(\frac{|\sqrt{(x-x_{\mathrm{c}})^{2}+(y-y_{\mathrm{c}})^{2}}-r|^{2}}{-2\sigma^{2}}\right) \end{cases}$$

where the upper function is for linear trench and the lower function is for arc trench.

The gradient $\nabla U(\mathbf{q})$ can be obtained as

$$\nabla U(\mathbf{q}) = \nabla U_{\mathrm{w}}(\mathbf{q}) + \nabla U_{\mathrm{t}}(\mathbf{q}). \tag{6.22}$$

6.3.3 Road Switching Strategy

The correct selection of road constraints has an effect on positioning performance. As mentioned earlier in Section 6.2.1.2, a scheme is proposed for road switching to find the most appropriate road constraint at all time. The switching strategy contains 1) switching condition, and 2) new constraint determination. As for arc constraints, switching is considered when the distance between sources of potential well and trench $d_{\mathrm{t}} > \epsilon$, where ϵ is a parameter defined according to constraint covariance σ, the density of road network and measurement accuracy. As for linear segments in most cases, we define the heading difference $\Delta\theta = |\theta_{\mathrm{rd}} - \theta_{\mathrm{m}}|$, where θ_{rd} denotes the heading of the road constraint, θ_{m} is the heading obtained from two consecutive measurement positions [129]. The road switching scheme is trigged once $\Delta\theta > 10^{\circ}$ and $d_{\mathrm{t}} > \epsilon$. After the switching condition is trigged, the arc constraint with the minimum d_{t} has priority to be the new constraint. Otherwise, the linear road segment around measurement with the minimum $\Delta\theta$ is selected as the new constraint.

The whole road switching process is demonstrated in Algorithm. 4.

Algorithm 4 Road Switching Strategy

Input: Linear candidates $\mathcal{C}^{\mathrm{s}} = \{c_1^{\mathrm{s}}(\mathbf{q}) = 0, \cdots, c_n^{\mathrm{s}}(\mathbf{q}) = 0\}$, arc candidates $\mathcal{C}_{\mathrm{arc}}^{\mathrm{s}} = \{c_{\mathrm{arc}1}^{\mathrm{s}}(\mathbf{q}) = 0, \cdots, c_{\mathrm{arc}m}^{\mathrm{s}}(\mathbf{q}) = 0\}$, current measurement $\mathbf{q}^{\mathrm{s}}$, current road $c_{\mathrm{bfr}}^{\mathrm{s}}$, heading difference $\Delta\theta$, distance between measurement and road segment d_{t}

Output: Road segment after switching $c_{\mathrm{aft}}^{\mathrm{s}}$

$i \leftarrow 1$
flag $\leftarrow 0$
$d_{\min} \leftarrow 50$
$\Delta\theta_{\min} \leftarrow 60$
if switching condition trigged **then**
  while $i \leq m$ **do**
    Calculate the distance $d_{\mathrm{arc}i}$ between $\mathbf{q}^{\mathrm{s}}$ and each arc in $\mathcal{C}_{\mathrm{arc}}^{\mathrm{s}}$
    if $d_{\mathrm{arc}i} < d_{\min}$ **then**
      $c_{\mathrm{aft}}^{\mathrm{s}} \leftarrow c_{\mathrm{arc}i}^{\mathrm{s}}(\mathbf{q}) = 0$
      flag $\leftarrow 1$
      $d_{\min} \leftarrow d_{\mathrm{arc}i}$
    end if
    $i \leftarrow i + 1$
  end while
  if $c_{\mathrm{aft}}^{\mathrm{s}}$ is not empty **then**
    goto 29
  end if
  $i \leftarrow 1$
  while $i \leq n$ **do**
    Calculate the heading difference $\Delta\theta_i$ for each segment in $\mathcal{C}^{\mathrm{s}}$
    Calculate the distance $d_{\mathrm{t}i}$ between $\mathbf{q}^{\mathrm{s}}$ and each segment in $\mathcal{C}^{\mathrm{s}}$
    if $d_{\mathrm{t}i} < 50$ and $\Delta\theta_i < \Delta\theta_{\min}$ **then**
      $c_{\mathrm{aft}}^{\mathrm{s}} \leftarrow c_i^{\mathrm{s}}(\mathbf{q}) = 0$
      flag $\leftarrow 0$
    end if
    $i \leftarrow i + 1$
  end while
end if
return $c_{\mathrm{aft}}^{\mathrm{s}}$

Table 6.1: Test sequences.

No.	Length (m)	Information source			Description
		VO	GPS	Road maps	
1	225.66	✓		✓	short path with turns
2	271.89	✓		✓	mostly straight path with backward moving and small turns
3	237.09	✓		✓	path with loops
4	594.17	✓		✓	newly-built road with an obsolete road map
5	1389.23	✓		✓	block loop closure
6	1326.50	✓	✓	✓	self-recorded path in low speed

6.4 Experimental Results

The summary of test sequences is listed in Table 6.1. The proposed fusion approach is tested with data provided by Karlsruhe Dataset (Sequence 1-3) [105], the Málaga Stereo and Laser Urban Data Set (Sequence 4-5) [130] and our data (Sequence 6). In the first five datasets, the true position is provided by a combined GPS/IMU system, and the information is fused according to road constraints and visual odometry data. In our data, the true position is provided by Trimble DSM12/212 DGPS. A low-cost GPS (ublox EVK-M8F), a visual odometry equipped with Bumblebee2 BB2-08S2C stereo camera and road constraints are combined to obtain estimated position. All self-generated road constraints in experiments are created from OpenStreetMap, while the map matching results are obtained from Google Maps Roads API.

6.4.1 Quantitative Results

The evaluations are undertaken with pure visual odometry (VO), map matching (MM) and potential field (ours) method, respectively. The performance of these methods are demonstrated in Table 6.2. As only 2D maps are provided in MM, vertical mean error and vertical standard deviation are not included in statistics. In experiments, the parameters in potential construction functions are set as $\alpha_t = 10$, $\alpha_w = 200$, trench covariance $\sigma = 20$, initial VO covariance is set as $\mathbf{D}_0 = \text{diag}(5, 5, 5)$

for Sequence 1-5 and $\mathbf{D}_0 = \text{diag}(20, 20, 20)$ for Sequence 6, with different increasing rate η from 1.005 to 1.01.

The evaluation indexes include total error, mean error and standard deviation. The total error is calculated from the difference of traveled distance compared to ground truth. The mean error and standard deviation are obtained by evaluating positioning results for each frame. The VO accuracy is relatively low in measuring straight movements and short distance, compared to turns, loops and larger scale environments. The performance of MM is influenced by vehicle's path: driving close to road center line will lead to smaller mean error, especially for wide roads, as most of the roads are modelled using center lines in digital maps. According to test results, the proposed fusion approach shows adaptability and robustness, as it achieves a good balance among evaluation indexes. The mean error and standard deviation are approaching the best results measured by various methods in different road conditions. It has to be pointed out that in Sequence 6, the large total error is due to the different sampling rate of DGPS and other sensors. The path obtained from DGPS reflects the actual moving trajectory of the vehicle, and it is not smooth as other paths. The DGPS measurements are down-sampled to 1 Hz to make comparisons with the fusion results.

6.4.2 Qualitative Evaluation

In this section, we mainly focus on two indexes of positioning results: small-scale sensitivity and large-scale drift. In order words, the movement in all directions should be detected in a short distance, while the drifting error should be reduced during a long run.

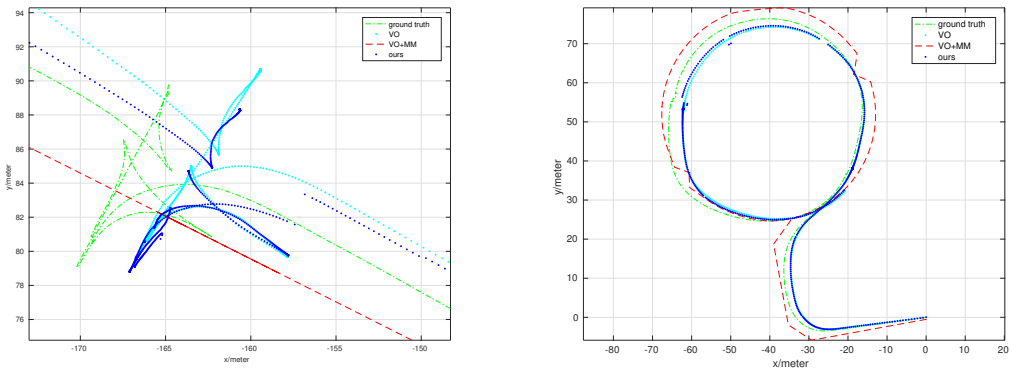
It is difficult for MM method to reflect local movement of the vehicle, especially in the vertical direction of the road segment, as the movement in this direction is regarded as invalid along the road. In Sequence 2, the vehicle reverses while traveling. As can

Table 6.2: Positioning results from visual odometry, map matching and the proposed approach.

No.	Method	Error		
		Total%	Mean$_{xy}$ (m)	Std$_{xy}$ (m)
1	VO	3.49	4.24	1.34
	VO+MM	11.35	3.68	1.97
	ours	6.90	3.48	1.36
2	VO	5.07	9.97	3.39
	VO+MM	14.90	10.47	3.37
	ours	8.42	9.20	3.11
3	VO	3.38	2.35	0.93
	VO+MM	8.86	3.42	1.13
	ours	2.86	2.10	1.22
4	VO	4.27	23.46	28.35
	VO+MM	10.16	34.71	16.84
	ours	4.88	23.21	16.87
5	VO	1.90	14.32	12.87
	VO+MM	3.17	11.91	9.12
	ours	3.76	12.14	9.65
6	VO	46.47	20.69	4.62
	GPS	39.55	8.58	9.44
	ours	40.21	7.67	5.55

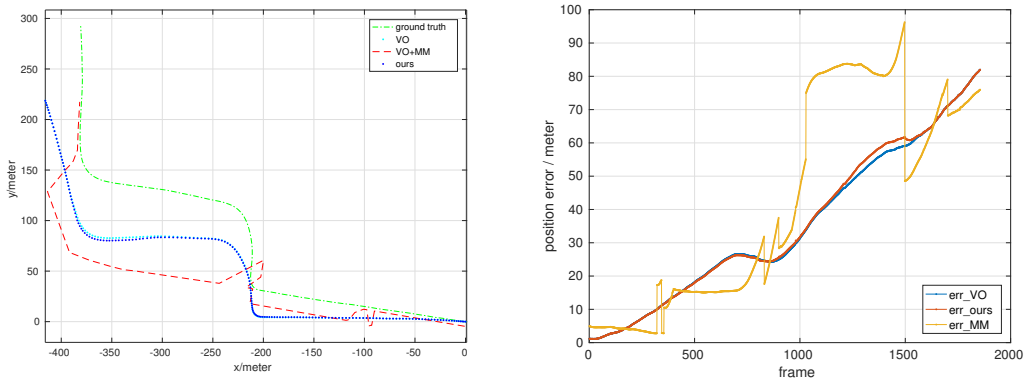
be seen in Fig. 6.2a, our method utilizes VO measurement in small-range movement, and the positioning error is bounded by restraining VO drifting. In Sequence 3, a loop trajectory is tested. Fig. 6.2b shows the poor continuity of MM result. The proposed method is able to balance between trajectory smoothness and localization mean error by adjusting covariances of potential functions. With the pace of urban construction, sometimes the digital maps are not updated promptly. In Sequence 4, the circumstance of driving along a newly built road is considered. Fig. 6.3a and 6.3b illustrate that our approach outperform MM when the vehicle is on a new path from frame 40 to 72, and it also reduces error compared to VO after frame 82.

Pure VO suffers from drifting problem along with distance. In Sequence 5, a block loop closure path is tested, with the results shown in Fig. 6.4a and Fig. 6.4b. The proposed approach produces an evidently robust result compared to pure VO method. The initial VO measurements with little drifting error tend to be trusted, while the fusion results are similar to map matched positions at the end. As shown



(a) Sequence 2. (b) Sequence 3.

Figure 6.2: Positioning results of Sequence 2 and 3. Trajectories of ground truth, visual odometry, map matching and the proposed potential field approach are plotted with different colors.

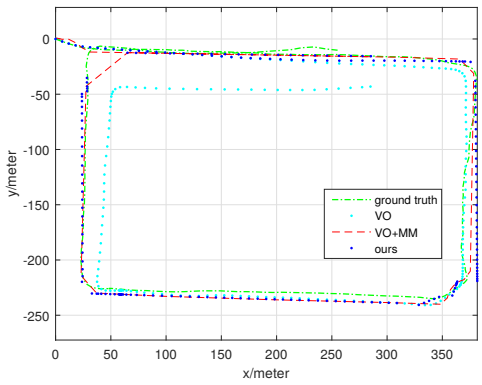


(a) Positioning results of Sequence 4. (b) Positioning error of Sequence 4.

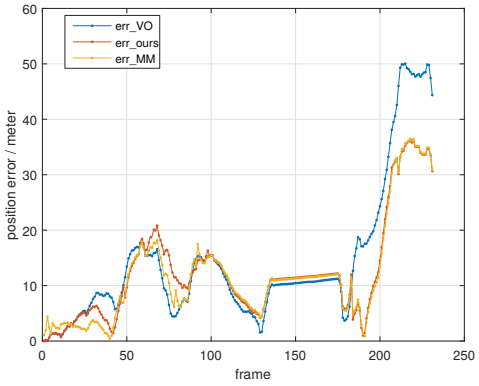
Figure 6.3: Trajectories of ground truth, visual odometry, map matching and the proposed potential field approach are plotted with different colors for Sequence 4. The corresponding positioning errors are depicted in the right figure.

in Fig. 6.5a and Fig. 6.5b, in Sequence 6, low-cost GPS provides unstable path, which deviates from ground truth when the vehicle is under urban canyon environments. Moreover, VO path drifts obviously such that only the geometry outline of the ground truth is presented. After the proposed fusion, the estimated results shows that positioning performance is greatly improved.

The proposed approach is implemented in Matlab and all the experiments are conducted on a mobile workstation with an i7-4710MQ/2.5GHz processor. It takes 16

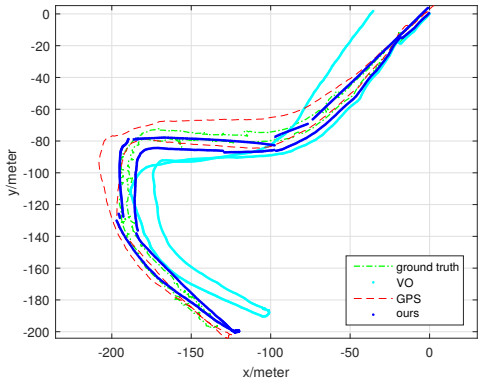


(a) Positioning results of Sequence 5.

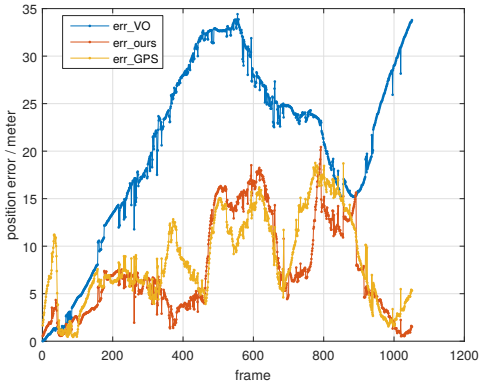


(b) Positioning error of Sequence 5.

Figure 6.4: Trajectories of ground truth, visual odometry, map matching and the proposed potential field approach are plotted with different colors for Sequence 5. The corresponding positioning errors are depicted in the right figure.



(a) Positioning results of Sequence 6.



(b) Positioning error of Sequence 6.

Figure 6.5: Trajectories of ground truth, visual odometry, map matching and the proposed potential field approach are plotted with different colors for Sequence 6. The corresponding positioning errors are depicted in the right figure.

ms per frame for minimum searching and 2 ms per frame for road switching.

There are some limitations of our work. As we have not take the dynamic model of the vehicle into consideration, the smoothness of the trajectory is not satisfying under specific conditions, which will have a negative effect on user experience even though the average positioning error is small. The unsmooth trajectory appears especially when the vehicle's path consists of short road segments. By creating the dynamic potential field, the dynamic model of the vehicle could be combined within

the fusion frame such that the trajectory smoothness could be improved.

6.5 Conclusion

In this chapter, we have presented a potential-field-based fusion approach for vehicle positioning with low-cost GPS, visual odometry, and digital map. By creating potential well and potential trench separately, the proposed approach combines multi-source information including position measurements and road constraints in an intuitive way, without additional map matching. The position is then estimated by searching for the minimum of the combined potential field. The experiment results show that the potential-field-based approach combines the advantages of multiple types of data, and is able to provide accurate and robust localization when high-precision GPS is not available. In the future, more efforts can be made for more types of sensors. Besides, we would like to explore the creation of dynamic potential field by considering system state equations and measurement equations as a part of the fusion frame.

Chapter 7

Conclusions and Future Work

This chapter concludes this thesis by summarizing the main contributions and suggesting directions for future research.

7.1 Summary of Contributions

Navigation based on visual sensors has drawn great attention due to its low cost and abundant information. In this thesis, we aim to develop robust vision based localization methods for UGV. Road-constrained metric localization approaches based on visual odometry have been presented, where visual odometry is in charge of motion update and road constraints from digital map are used to correct the accumulated error of visual odometry. Topological localization approach based on place recognition has also been developed. A hybrid approach which combines topological and metric localization has been proposed to play their respective advantages. To deal with situations where multiple information sources are available, a sensor fusion approach is presented to localize vehicles. All the approaches developed in this thesis are applicable to UGV navigation in GPS challenging environments. To demonstrate the performance, results for vehicle localization in a number of large

scale environment have been presented.

The main contributions and conclusions of this thesis are summarized as follows:

1. A metric localization approach based on the fusion of visual odometry and road constraints (**Chapter 3**).

   Drift and scale ambiguity are two main issues which reduce localization accuracy in monocular visual odometry. It is necessary to propose a unified model to represent these measurement uncertainties. In **Chapter 3**, we have presented a localization framework to globally localize a mobile vehicle equipped with monocular camera and a freely available digital map. In this framework, no other sensor is involved to tackle the accumulated drift and scale ambiguity. Inspired by the concept of cloud, a Gaussian-Gaussian Cloud model has been proposed. The drift and scale ambiguity of monocular visual odomtry are both considered as measurement uncertainties and incorporated into the presented Gaussian-Gaussian Cloud model. A shape matching scheme which evaluates the alignment of different trajectories to the digital map, has been used to filter out cloud drops which are inconsistent with road constraints. Based on the statistical properties of Gaussian-Gaussian Cloud model, a parameter estimation scheme has been implemented to narrow down the scale ambiguity while resampling cloud drops. Comprehensive evaluations with practical data have been conducted to verify the effectiveness of the proposed localization framework. Experiment results have shown that with road constraints, the localization error has been significantly reduced compared to the advanced method.

2. A topological localization approach based on place recognition (**Chapter 4**).

   Most of the current place recognition methods are designed for the application in a specific environment. Place recognition method which is applicable to various environments is very desired. Point features suffer from illumination variation, while line features suffer from position ambiguity. Point and line

features have different expression capabilities in different environments. A good combination of point and line features seems to improve the robustness and accuracy of place recognition. In **Chapter 4**, a modified vocabulary tree that can combine multiple feature types for place recognition has been proposed. The ability of combining different feature types makes it possible for users to customize feature combination for various environments. Experiment results on real-world datasets have demonstrated the advantage of our system compared with existing approaches. Although only the combination of point and line features has been discussed in this study, the combination of other feature types is also promising.

3. A metric-topological localization approach based on the integration of place recognition, visual odometry and road constraints (**Chapter 5**).

   Although good localization results have been obtained from the road constrained method presented in **Chapter 3**, the following issues can not be ignored. Firstly, the road constrained approach only works in on-road scenarios due to the on-road assumption. Secondly, the vehicle's initial position and orientation are unknown and the initialization process relies extremely on shape matching performance. If we look at place recognition approaches, they do not suffer from the above issues. Despite the fact that the metric position of the query image can not be computed, a rough topological location can be obtained through place recognition operation no matter the vehicle is on road or off road. On the other hand, a rough position information helps significantly to the initialization process of the road constrained approach. Hence, it is promising to incorporate place recognition into our road constrained approach.

   In **Chapter 5**, an integrated strategy has been proposed to localize a mobile vehicle equipped with one panoramic camera, one mono-camera and one digital map. Place recognition, visual odometry and road-constrained approaches have been incorporated into one framework. With in this framework, an on-

road/off-road judging scheme has been proposed such that the integrated approach is applicable for both on road and off road scenarios. If the judging scheme gives on-road predication, the full road-constrained approach will be applied; otherwise visual odometry takes over. Place recognition plays a role of topological localization and assists initialization process of the whole framework. The time consumption of initialization process has been significantly reduced. Besides, A detailed comparison between Gaussian and Uniform assumption modelling the scale distribution of monocular visual odometry has been discussed. Moreover, a mutual check thread has been implemented to give a criterion for judging whether metric and topological results are sufficiently consistent. Evaluation results show that the proposed framework is highly accurate.

4. A metric localization approach based on the fusion of visual odometry, low-cost GPS and digital map (**Chapter 6**).

   In previous chapters, we focus on the development of vision-based localization without the participation of GPS. The geometric shapes of road networks are considered as constraints to assist with position estimation. And shape matching method is utilized to evaluate the alignment of different trajectories to the digital map. However, in real application, low-cost GPS is available from time to time. The performance of shape matching is affected by map resolution. In **Chapter 6**, we have proposed a fusion approach to localize urban vehicles with the participation of low-cost GPS, visual odometry and digital map. Distinguished from conventional localization methods, the concept of artificial potential field that is widely used for obstacle avoidance has been presented to represent measurements and constraints, respectively. Position measurements from visual odometry and low-cost GPS are modelled with a potential well function, while road constraints from digital map are modelled with a potential trench function without additional map matching. By searching for the minimum of the combined potential field, the position can be estimated. Ex-

periments conducted on real world datasets have demonstrated the robustness and high positioning accuracy of the proposed potential field based approach.

7.2 Recommendations for Future Work

Although comprehensive studies on vision-based localization have been carried out in this thesis, a lot of work still needs to be done to improve the localization performance. The following session lists our future work:

1. The method proposed in **Chapter 4** is a heuristic mixing of point and line features and only tested on a relative small dataset. Bigger dataset as well as the combination of other feature types will be tested in the future research. Real-time process is very important for mobile vehicles. However, no experiments on the computation increase caused by the combination of point and line features have been done. Thus, the computation analyses also need to be performed in the near feature. For all the experiments carried out in **Chapter 4**, no geometrical or temporal consistent checking is applied to improve the performance. In the future, we plan to use these verification approaches and extend this work to a visual SLAM system.

2. As concluded in **Chapter 1**, it is promising to conduct Convolutional Neural Networks based place recognition in modern application. Generally speaking, there are two kinds of CNN-based place recognition. The first category is similar to conventional place recognition except that the feature expressing a query image is learned from a CNN model. A classification step is required to find the best matched image from the geo-tagged database. Hence, it is still a classification process. To the contrary, the second category is considered as a regression process. Given a query image, the camera's 6 DOF pose relative to a scene is regressed from a convolutional neural network trained end-to-end. For the next step, we intend to develop the two types of CNN-based

place recognition approach. Google Street View provides 360° view images on major roads all of the world. Such information can be used to form the huge training dataset.

3. In **Chapter 3, 5**, and **6**, the open licensed OpenStreetMap is used as our digital map. It is freely available and easily editable. However, it has several downsides. On one hand, the uncertainty of the map is not considered while generating the mathematical model. During our experiment, we found that the map is quite accurate at urban area, where the map is frequently updated; but less accurate at sparsely populated region. On the other hand, as all geometric maps are modelled with the road center line, there is always an offset between vehicles actual trajectory and map-represented roads. The offset is positively relative to road width and can deliver positioning error in transverse direction. Thus, either a map uncertainty model or a High Precision Map which can provide lane level positioning accuracy is desired.

Author's Publications

Journal Papers:

1. **S. Yang**, R. Jiang, H. Wang, and S. S. Ge, "Road Constrained Monocular Visual Localization Using Gaussian-Gaussian Cloud Model," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2017.2685436, 2017, Accepted.

2. **S. Yang**, W. Mou, H. Wang (2015) Place Recognition using Multiple Feature Types. Adv Robot Autom S2:008. doi: 10.4172/2168-9695.S2-008, 2015, Accepted.

3. R. Jiang, **S. Yang**, S. S. Ge, H. Wang and T. H. Lee, "Geometric Map-Assisted Localization for Mobile Robots Based on Uniform-Gaussian Distribution," in IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 789-795, April 2017. (Co-first author)

4. R. Jiang, **S. Yang**, S. S. Ge, X. M. Liu, H. Wang and T. H. Lee, "GPS/Odometry /Map Fusion for Vehicle Positioning Using Potential Function," in Autonomous Robots, doi 10.1007/s10514-017-9646-9, 2017, Accepted.

5. H. Wang, W. Mou, X. Mou, S. Yuan, S. Ulun, **S. Yang**, and B.-S. Shin, "An automatic self-calibration approach for wide baseline stereo cameras using sea surface images," Unmanned Systems, vol. 3, no. 4, pp. 277290, 2015.

Conference Papers:

1. **S. Yang**, W. Mou, H. Wang and S. S. Ge, "Place recognition by combining multiple feature types with a modified vocabulary tree," 2015 International Conference on Image and Vision Computing New Zealand (IVCNZ), Auckland, 2015, pp. 1-6, Accepted.

2. **S. Yang**, R. Jiang, H. Wang and S. S. Ge, "Integrated Metric-topological Localization by Fusing Visual Odometry, Digital Map and Place Recognition," 2017, International Conference on Man-Machine Interactions, Cracow, Poland, Accepted.

3. **S. Yang** and H. Wang, "Improving Vision-based Topological Localization by Combining Local and Global Image Features," 7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, PPNIV 2015, IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, Germany, September 28, 2015, Accepted.

4. H. Wang, X. Mou, W. Mou, S. Yuan, S. Ulun, **S. Yang**, and B.-S. Shin, "Vision based long range object detection and tracking for unmanned surface vehicle," in IEEE International Conference on Robotics, Automation and Mechatronics, Angkor Wat, Cambodia, 2015, pp. 101-105, Accepted.

Bibliography

[1] T. Kos, I. Markezic, and J. Pokrajcic, "Effects of multipath reception on gps positioning performance," in *Elmar, 2010 Proceedings.* IEEE, 2010, pp. 399–402.

[2] E. Guizzo, "How googles self-driving car works," *IEEE Spectrum Online, October*, vol. 18, 2011.

[3] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, "Imu-based localization and slip estimation for skid-steered mobile robots," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 2845–2850.

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[5] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, June 2004, pp. I–652–I–659 Vol.1.

[6] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

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

[8] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1015–1026, Oct 2008.

[9] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[10] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 298–304.

[11] S. Wirth, P. L. N. Carrasco, and G. O. Codina, "Visual odometry for autonomous underwater vehicles," in *2013 MTS/IEEE OCEANS - Bergen*, June 2013, pp. 1–6.

[12] Y. Cheng, M. W. Maimone, and L. Matthies, "Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 54–62, June 2006.

[13] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[14] J.-M. Lavest, M. Viala, and M. Dhome, "Do we really need an accurate calibration pattern to achieve a reliable camera calibration?" in *European Conference on Computer Vision*. Springer, 1998, pp. 158–174.

[15] A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald, "Using the kinect as a navigation sensor for mobile robotics," in *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*. ACM, 2012, pp. 509–514.

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

[17] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[18] J. Zhang, M. Kaess, and S. Singh, "Real-time depth enhanced monocular odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014.

[19] B. M. Kitt, J. Rehder, A. D. Chambers, M. Schonbein, H. Lategahn, and S. Singh, "Monocular visual odometry using a planar road model to solve scale ambiguity," 2011.

[20] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1566–1573.

[21] D. Zhou, Y. Dai, and H. Li, "Reliable scale estimation and correction for monocular visual odometry," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 490–495.

[22] B. Lee, K. Daniilidis, and D. D. Lee, "Online self-supervised monocular visual odometry for ground vehicles," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5232–5238.

[23] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on.* IEEE, 2003, pp. 1403–1410.

[24] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

[25] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," Sydney, Australia, December 2013.

[26] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[27] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1246–1257, June 2014.

[28] V. Vineet, O. Miksik, M. Lidegaard, M. Niener, S. Golodetz, V. A. Prisacariu, O. Khler, D. W. Murray, S. Izadi, P. Prez, and P. H. S. Torr, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 75–82.

[29] B. Kitt, A. Geiger, and H. Lategahn, "Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme," in *Intelligent Vehicles Symposium (IV)*, 2010.

[30] I. Cvii and I. Petrovi, "Stereo odometry based on careful feature selection and tracking," in *2015 European Conference on Mobile Robots (ECMR)*, Sept 2015, pp. 1–6.

[31] A. Desai and D. J. Lee, "Visual odometry drift reduction using syba descriptor and feature transformation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1839–1851, July 2016.

[32] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct 2015.

[33] J. Cheng, C. Leng, J. Wu, H. Cui, and H. Lu, "Fast and accurate image matching with cascade hashing for 3d reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[34] H. Badino, A. Yamamoto, and T. Kanade, "Visual odometry by multi-frame feature integration," in *International Workshop on Computer Vision for Autonomous Driving @ ICCV*, December 2013.

[35] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[36] A. Ranganathan, "The levenberg-marquardt algorithm," *Tutoral on LM algorithm*, pp. 1–5, 2004.

[37] R. Raguram, J. M. Frahm, and M. Pollefeys, "Exploiting uncertainty in random sample consensus," in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 2074–2081.

[38] F. A. Moreno, J. L. Blanco, and J. Gonzlez-Jimnez, "Erode: An efficient and robust outlier detector and its application to stereovisual odometry," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 4691–4697.

[39] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2521–2526.

[40] M. Buczko and V. Willert, "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 478–483.

[41] I. Kostavelis, E. Boukas, L. Nalpantidis, and A. Gasteratos, "Visual odometry for autonomous robot navigation through efficient outlier rejection," in *2013 IEEE International Conference on Imaging Systems and Techniques (IST)*, Oct 2013, pp. 45–50.

[42] N. Sünderhauf, K. Konolige, S. Lacroix, and P. Protzel, "Visual odometry using sparse bundle adjustment on an autonomous outdoor vehicle," in *Autonome Mobile Systeme 2005*. Springer, 2006, pp. 157–163.

[43] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," in *Robotics research*. Springer, 2010, pp. 201–212.

[44] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments," in *the 12th International Symposium on Experimental Robotics (ISER)*, vol. 20. Citeseer, 2010, pp. 22–25.

[45] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras," in *Robot Soccer World Cup*. Springer, 2011, pp. 306–317.

[46] I. Dryanovski, R. G. Valenti, and J. Xiao, "Fast visual odometry and mapping from rgb-d data," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2305–2310.

[47] R. G. Valenti, I. Dryanovski, C. Jaramillo, D. P. Strm, and J. Xiao, "Autonomous quadrotor flight using onboard rgb-d visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5233–5238.

[48] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[49] Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3934–3942.

[50] C. Raposo, M. Lourenço, M. Antunes, and J. P. Barreto, "Plane-based odometry using an rgb-d camera." in *BMVC*, 2013.

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

[52] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[53] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2100–2106.

[54] D. Li and Y. Du, *Artificial intelligence with uncertainty*. CRC press, 2007.

[55] D. Li, C. Liu, and W. Gan, "A new cognitive model: cloud model," *International Journal of Intelligent Systems*, vol. 24, no. 3, pp. 357–375, 2009.

[56] G.-Y. Hu and P.-L. Qiao, "Cloud belief rule base model for network security situation prediction," *IEEE Communications Letters*, vol. 20, no. 5, pp. 914–917, 2016.

[57] X. Sun, C. Cai, and X. Shen, "A new cloud model based human-machine cooperative path planning method," *Journal of Intelligent & Robotic Systems*, vol. 79, no. 1, pp. 3–19, 2015.

[58] A. Kavousi-Fard, T. Niknam, and M. Fotuhi-Firuzabad, "A novel stochastic framework based on cloud theory and *theta* -modified bat algorithm to solve the distribution feeder reconfiguration," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 740–750, March 2016.

[59] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models." in *IJCAI*, vol. 7, no. 1, 2007, pp. 2480–2485.

[60] D. M. Cole and P. M. Newman, "Using laser range data for 3d slam in outdoor environments," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1556–1563.

[61] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, p. 263, 2008.

[62] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[63] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Aaai/iaai*, 2002, pp. 593–598.

[64] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 3562–3568.

[65] D. Törnqvist, T. B. Schön, R. Karlsson, and F. Gustafsson, "Particle filter slam with high dimensional vehicle model," *Journal of Intelligent & Robotic Systems*, vol. 55, no. 4, pp. 249–266, 2009.

[66] H. Strasdat, A. J. Davison, J. M. M. Montiel, and K. Konolige, "Double window optimisation for constant time visual slam," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2352–2359.

[67] M. Fanfani, F. Bellavia, and C. Colombo, "Accurate keyframe selection and keypoint tracking for robust visual odometry," *Machine Vision and Applications*, vol. 27, no. 6, pp. 833–844, 2016.

[68] H. Lim, J. Lim, and H. J. Kim, "Real-time 6-dof monocular visual slam in a large-scale environment," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 1532–1539.

[69] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual slam: why filter?" *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[70] A. R. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *European Conference on Computer Vision*. Springer, 2010, pp. 255–268.

[71] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "Mav urban localization from google street view data," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3979–3986.

[72] G. Floros, B. van der Zander, and B. Leibe, "Openstreetslam: Global vehicle localization using openstreetmaps," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1054–1059.

[73] N. Mattern, R. Schubert, and G. Wanielik, "High-accurate vehicle localization using digital maps and coherency images," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*. IEEE, 2010, pp. 462–469.

[74] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with gps in urban canyon environments," *IEEE transactions on robotics and automation*, vol. 19, no. 1, pp. 15–25, 2003.

[75] X. Zhang, Q. Wang, and D. Wan, "Map matching in road crossings of urban canyons based on road traverses and linear heading-change model," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, no. 6, pp. 2795–2803, 2007.

[76] G. Taylor and G. Blewitt, "GPS-GIS map matching: Combined positioning solution," *Intelligent Positioning: GIS-GPS Unification*, pp. 97–113, 2006.

[77] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2009, pp. 336–343.

[78] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[79] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[80] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8.

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

[82] D. Galvez-Lopez and J. D. Tardos, "Real-time loop detection with bags of binary words," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, sept. 2011, pp. 51 –58.

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

[84] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Computer vision–ECCV 2006*, pp. 430–443, 2006.

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

[86] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. Ieee, 2006, pp. 2161–2168.

[87] J. H. Lee, S. Lee, G. Zhang, J. Lim, W. K. Chung, and I. H. Suh, "Outdoor place recognition in urban environments using straight lines," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 5550–5557.

[88] M. J. Milford and G. 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. IEEE, 2012, pp. 1643–1649.

[89] H. Lategahn, J. Beck, B. Kitt, and C. Stiller, "How to learn an illumination robust image feature for place recognition," in *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.

[90] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *arXiv preprint arXiv:1504.06066*, 2015.

[91] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," June 2015.

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

[93] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *British Machine Vision Conference*, 2014.

[94] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European conference on computer vision.* Springer, 2014, pp. 584–599.

[95] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014, pp. 157–166.

[96] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik, "Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation," *International Journal of Computer Vision (IJCV)*, 2014.

[97] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake, "Efficient human pose estimation from single depth images," *Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, 2013.

[98] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *arXiv preprint arXiv:1411.1509*, 2014.

[99] Y. Hou, H. Zhang, and S. Zhou, "Convolutional neural network-based image representation for visual loop closure detection," in *Information and Automation, 2015 IEEE International Conference on.* IEEE, 2015, pp. 2238–2245.

[100] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[101] T. Botterill, S. Mills, and R. Green, "Correcting scale drift by object recognition in single-camera slam," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1767–1780, Dec 2013.

[102] M. A. Brubaker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3057–3064.

[103] V. K. Rohatgi and A. M. E. Saleh, *An introduction to probability and statistics.* John Wiley & Sons, 2015.

[104] M. Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1696–1703.

[105] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[106] H. Wang, W. Mou, H. Suratno, G. Seet, M. Li, M. Lau, and D. Wang, "Visual odometry using rgb-d camera on ceiling vision," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, Dec 2012, pp. 710–714.

[107] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on lbd descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[108] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 4, pp. 722–732, April 2010.

[109] R. Jiang, S. Yang, S. S. Ge, H. Wang, and T. H. Lee, "Geometric map-assisted localization for mobile robots based on uniform-gaussian distribution," *IEEE Robotics and Automation Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[110] H. Lategahn, J. Beck, B. Kitt, and C. Stiller, "How to learn an illumination robust image feature for place recognition," in *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia, 2013.

[111] P. Huang and Y. Pi, "Urban environment solutions to gps signal near-far effect," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 26, no. 5, pp. 18–27, 2011.

[112] S.-H. Kong, "Statistical analysis of urban gps multipaths and pseudo-range measurement errors," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, no. 2, pp. 1101–1113, 2011.

[113] J. Borenstein and L. Feng, "Measurement and correction of systematic odometry errors in mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 6, pp. 869–880, 1996.

[114] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 3, pp. 328–342, 1995.

[115] S. Se, D. G. Lowe, and J. J. Little, "Vision-based global localization and mapping for mobile robots," *IEEE Transactions on robotics*, vol. 21, no. 3, pp. 364–375, 2005.

[116] A. Ramisa, A. Tapus, D. Aldavert, R. Toledo, and R. Lopez de Mantaras, "Robust vision-based robot localization using combinations of local feature region detectors," *Autonomous Robots*, vol. 27, no. 4, pp. 373–385, 2009.

[117] I. P. Alonso, D. F. Llorca, M. Gavilán, S. Á. Pardo, M. Á. García-Garrido, L. Vlacic, and M. Á. Sotelo, "Accurate global localization using visual odometry and digital maps on urban environments," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 4, pp. 1535–1545, 2012.

[118] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 12, pp. 1696–1710, 2006.

[119] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of mems-based ins/gps integration for low-cost navigation applications," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 3, pp. 1077–1096, 2009.

[120] L. Wei, C. Cappelle, and Y. Ruichek, "Camera/laser/gps fusion method for vehicle positioning under extended nis-based sensor validation," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 11, pp. 3110–3122, 2013.

[121] J. L. Crassidis, "Sigma-point kalman filtering for integrated gps and inertial navigation," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, no. 2, pp. 750–756, 2006.

[122] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 329–343, 2012.

[123] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[124] S. S. Ge and C.-H. Fua, "Queues and artificial potential trenches for multirobot formations," *Robotics, IEEE Transactions on*, vol. 21, no. 4, pp. 646–656, 2005.

[125] D. Hall and J. Llinas, *Multisensor data fusion*. CRC press, 2001.

[126] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[127] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[128] R. Jiang, R. Klette, and S. Wang, "Modeling of unbounded long-range drift in visual odometry," in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*. IEEE, 2010, pp. 121–126.

[129] G. Jagadeesh, T. Srikanthan, and X. Zhang, "A map matching method for gps based real-time vehicle location," *Journal of Navigation*, vol. 57, no. 03, pp. 429–440, 2004.

[130] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.