Towards Long-Term Vision-Based Localization in Support of Monocular Visual Teach and Repeat

by

Nan Zhang

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto

Abstract

Towards Long-Term Vision-Based Localization in Support of Monocular Visual Teach and Repeat

Nan Zhang
Master of Applied Science
University of Toronto Institute for Aerospace Studies
University of Toronto
2018

This thesis presents an unsupervised learning framework within the Visual Teach and Repeat system to enable improved localization performance in the presence of lighting and scene changes. The resulting place-and-time-dependent binary descriptor is able to be updated as new experiences are gathered. We hypothesize that adapting the description function to a specific environment will improve the localization performance and allow the system to operate for a longer period of time before localization failure.

We also present a low-cost monocular Visual Teach and Repeat system, which uses a calibrated camera and wheel odometry measurements for navigation in both indoor and outdoor environments. These two parts are then combined with the end goal of achieving a low-cost, robust, and easily deployable system that enables navigation in complex indoor and outdoor environments with the eventual goal of long-term operation.

Acknowledgements

This thesis was completed with the help and support and many individuals, without whom this would not have been possible. I would like to express my gratitude towards all of them in helping me reach this point.

Foremost, I would like to thank Dr. Timothy D. Barfoot, my supervisor and mentor throughout my time at UTIAS. His guidance and advice were extremely valuable over the course of the two years.

I would also like to thank Dr. Michael Warren for his co-supervision on the various projects. He was always available to help when a problem arose and provided his expertise.

I would like to thank everyone at the Autonomous Space Robotics Lab for their help with debugging code, proofreading papers, setting up robots, and just being there in times of need. You are all an inspiration, the best of friends and amazing colleagues.

Finally, I would like to thank my family for their support, without whom I might have never started this degree.

This work was completed with funding from Ontario Graduate Scholarship (OGS) and Centre for Aerial Robotics Research and Education (CARRE).

Nan Zhang

Contents

List of Tables

List of Figures

Acrynoms

UAVs Unmanned Aerial Vehicles

VT&R Visual Teach and Repeat

IMU Inertial Measurement Unit

LiDAR Light Detection And Ranging

MAV Micro Aerial Vehicle

MLESAC Maximum Likelihood Estimation SAmple Consensus

RANSAC RANdom SAmple Consensus

GRIC Geometric Robust Information Criterion

SLAM Simultaneous Localization and Mapping

STPG Spatio-Temporal Pose Graph

STEAM Simultaneous Trajectory Estimation And Mapping

MPC Model Predictive Control

PnP Perspective-Three-Point

PTAM Parallel Tracking and Mapping

SURF Speeded-Up Robust Features

VO Visual Odometry

SVO Semi-Direct Visual Odometry

ORB Oriented-Rotated BRIEF

BRIEF Binary Robust Independent Elementary Features

SIFT Scale-Invariant Feature Transform

CPU Central Processing Unit

UTIAS University of Toronto Institute of Aerospace Studies

LIFT Learned Invariant Feature Transform

EKF Extended Kalman Filter

MAP maximum a posteriori

SfM structure from motion

SSD sum of squared differences

BoW Bag of Words

DTAM Dence Tracking and Mapping

AR Augmented Reality

VR Virtual Reality

LSD Large Scale Direct

FabMap Fast Appearance Based Mapping

SP super pixel

ACP apearance change prediction

SeqSLAM Sequence SLAM

YOLO You Only Look Once

SSD Single Shot Detector

R-CNN Region-based Convolutional Neural Network

Notation

a	Symbols in this font are real scalars.
$\mathbf{a}$	Symbols in this font are real column vectors.
$\mathbf{A}$	Symbols in this font are real matrices.
$(\cdot)_k$	The value of a quantity at timestep k.
$g(\cdot)$	A function g
$\boldsymbol{\mathcal{F}}_{\underrightarrow{a}}$	A vectrix representing a reference frame in three dimensions.
$\underrightarrow{a}$	A vector quantity in three dimensions.
$\mathbf{1}$	The identity matrix.
$\mathbf{0}$	The zero matrix.
$\boldsymbol{p}_j$	Symbols in this font are homogenous vectors with the last element being a 1 and the rest representing a point.
$\mathbf{p}_j$	Symbols in this font are non-homogenous vectors representing a point.

Chapter 1

Introduction

1.1 Motivation

Vision-based robotic navigation systems have achieved significant results in recent years, demonstrating the ability to safely navigate a variety of vehicles over longer distances in increasingly complex environments. There are two major factors preventing the wide adoption of such systems. The first is hardware complexity and cost. The second is the long-term performance of the system under natural scene changes.

This work seeks to address these two major issues by reducing the hardware requirements to a single calibrated monocular camera and a computer with a multi-core Central Processing Unit (CPU). Monocular cameras are ubiquitous, present on most cellphones and laptops being produced today. This demand has enabled the sensor to shrink in both size and cost. In addition to the low cost associated with a monocular solution, it is also deployable in a wider range of scenarios compared to a stereo solution. For example, in the case of high altitude Unmanned Aerial Vehicles (UAVs), a stereo solution is not physically feasible due to the large baseline distance required.

To address the second issue of long-term operation, different methods of vision-based localization are examined. With the success of machine learning, especially in the domain

of computer vision, we propose a learning-based approach within the Visual Teach and Repeat (VT&R) system to improve the localization performance using previous experiences as training data in an automated fashion.

As the name suggests, there are two components to VT&R: teach and repeat. During the teach phase, a user commands a vehicle through the environment. A map is built using stereo visual odometry (VO) and stored in the form of a spatial-temporal pose graph [45]. Windowed bundle adjustment (BA) is performed periodically to optimize the landmark positions and vehicle positions. Each vertex in the graph corresponds to a keyframe containing all the observed landmarks. The edges contain the estimated transformations and uncertainties between vertices. During the repeat phase, newly observed landmarks are matched against the map and passed through random sample consensus (RANSAC) to obtain a pose estimate. A path tracking controller then minimizes the cross-track error between the current pose and the closest vertex in the map. This allows the vehicle to follow the originally taught path both forwards and backwards.

1.2 Objective

The primary research objective deals with the long-term operational aspect of the system. Under a vision-based localization framework, the system must be able to deal with the dynamic and noisy nature of the real world. As a step towards a life-long localization system, we wish to extend the duration of time a robot is able to autonomously navigate in an environment using VT&R before localization failure. There are several approaches to this problem:

1. Use more images (multi-channel, multi-experience)
2. Use the entire image (dense methods)
3. Use depth and geometry information (RGB-D, stereo)
4. Use semantic information (identify high level abstractions)
5. Improve landmark correspondences in the presence of scene changes

The first three methods all require more information to be gathered or processed from the environment. The multi-channel approach aggregates information from multiple sources. This ensures some redundancy and can be useful when a certain viewpoint is not visually rich in information. Different types of preprocessing of images can also be done under the multi-channel approach. Each type of preprocessing can be optimized for specific environmental conditions.

Dense methods use all the pixels in the image instead of only the visually distinct areas. They also require depth information in order to construct a 3D model of the environment. The use of semantic information is an area of research only being investigated recently. The techniques under this class of approaches are still relatively new and still being developed. The dominant approaches include identifying high-level features in the environment and down-weighting poor or ambiguous features.

We focus on the correspondence problem. This approach easily extends the existing feature-based localization framework of VT&R and we can leverage the rich data source generated from VT&R for automatically generating labeled correspondences unique to the particular environment. Using a learning approach, the various experiences (traversals) over a path can be used to incrementally learn better description functions that improve the landmark correspondences. This results in a place-and-time-dependent descriptor that adapts to different environmental conditions.

1.3 Contributions

This work extends the existing VT&R 2.0 framework. The first contribution is the experimental validation of the monocular VT&R system working in-the-loop on a ground vehicle without any assumptions regarding scene geometry [39]. The second component is the development of a learning-based descriptor which improves the localization performance in the presence of illumination and seasonal changes [40]. Finally, we show the entire monocular system

working in-the-loop with the adaptive place-and-time-dependent descriptors. The end result demonstrates:

- The feasibility of a low-cost monocular Visual Teach and Repeat system that makes no assumptions about scene geometry
- Improved localization performance using the proposed place-and-time-dependent descriptors compared to traditional descriptors
- Experimental demonstration of the entire navigation and learning system working in-the-loop

1.4 Thesis Overview

This remainder of this thesis is split into five main chapters. In chapter 2, a summary of the current state of the art techniques in the field of vision-based navigation is presented, with a focus on the monocular specific techniques. Different approaches to enable long-term vision-based localization are also examined.

In Chapter 3, the core VT&R system is presented as background to the work presented. Relevant notation and the core mathematical background is also explained.

In Chapter 4 the modifications required to enable monocular operation are highlighted along with experimental results of the system working in the loop on a complex terrain in the University of Toronto Institute of Aerospace Studies (UTIAS) MarsDome. Limitations and points of failure are discussed along with possible solutions.

In Chapter 5, a learning approach is presented as a method to enable long-term localization. We explore learning binary descriptor using an evolutionary algorithm and present the results using offline analysis on two datasets.

Finally, Chapter 6 combines the monocular VT&R system with the learned place-and-time-dependent descriptors as a demonstration of the fully integrated system.

Chapter 2

Literature Review

2.1 SLAM Systems

2.1.1 History

According to [15], the first conception of the probabilistic Simultaneous Localization and Mapping (SLAM) problem was in 1986 at the IEEE Robotics and Automation Conference. The central idea was to estimate the spatial relationships between stationary landmarks and the robot positions as the robot moved through an environment. A breakthrough occurred around 1995 when it was proven that the problem is actually convergent, contrary to what many researchers thought. In other words, regardless of the motion of the robot, the estimated relative landmark positions converged. Many different formulations of the problem arose soon after using Extended Kalman Filter (EKF), Particle Filters and maximum a posteriori (MAP) estimation. More details on the various formulations can be found in Probabilistic Robotics [61] and State Estimation for Robotics [3].

SLAM is formulated to use any type of observation model, hence a variety of sensors can be used in combination to estimate landmark and robot positions. Visual-SLAM specifically focuses on using a camera as the only sensor. There is significant overlap between the photogrammetry field and visual-SLAM. Some examples include the bundle adjustment problem

(explained later) and structure from motion (SfM). More details on photogrammetry concepts can be found in Multiple View Geometry in Computer Vision [20].

Traditionally, feature-based methods are used for Visual-SLAM with each point feature in the camera image corresponding to a landmark. This relies upon a detection and description function to establish data correspondence and result in a point cloud of sparse landmarks in the environment [24, 43]. Dense methods, on the other hand, use the entire image and result in semi-dense 3D maps of the environment [16,64]. Other hybrid approaches such as Semi-Direct Visual Odometry (SVO) have also been explored [18].

Dense methods are still relatively a new field and require much more computational power compared to feature-based methods. As online, long-term operation onboard a robotic vehicle is a critical requirement, we continue with the proven approach of feature-based methods due to their robustness and speed over the dense methods.

A full SLAM system is able to simultaneously map and localize without any prior knowledge of the environment. Loop closure and place recognition play an important part in this process. By automatically recognizing a previously visited location, the new observations can be used to update all the state estimates resulting in a much more metrically accurate map. In this sense VT&R is a not a fully SLAM system as it only performs mapping and localization. Instead, the system leverages the human operator for place-recognition tasks.

2.1.2 Monocular SLAM Systems

Recently, monocular systems have received a significant amount of interest due to the ubiquity of individual cameras in smart-phones and computers. This is contrasted to stereo systems which require tight tolerances on the hardware and enough space to accommodate the camera. The literature for Visual-SLAM can be loosely categorized into two primary approaches: better established sparse (feature-based) methods [11,24,25] and more recent dense methods [16,64]. Learning-based methods have also proven extremely successful at estimating depth from a single image [59].

Some vision-based systems rely on stereo, or a combination of monocular and inertial sensors to achieve robustness and ensure a certain degree of scale accuracy. Stereo systems are limited by a rigid baseline, putting a hard constraint on the distance of features that can be triangulated. Monocular visual-inertial systems [17] achieve significant accuracy with only a monocular camera but are significantly more complex and costly. Cheaper Inertial Measurement Unit (IMU) are extremely noisy and can only be relied upon for very short distances before significant drift is observed. This means the IMU is only able to be used for approximate scale recovery. A more expensive IMU allows for a tightly coupled system but then we get the complexity involved in properly calibrating the system.

An example of early work using an upward facing monocular camera is the MINERVA museum robot which uses both laser scans and monocular images to localize itself to an occupancy grid map using a Monte-Carlo or particle filtering approach [60]. In [22], in addition to Scale-Invariant Feature Transform (SIFT) features, they also use corners, light sources and door frames as features for navigating in indoor environments.

Some more recent feature-based SLAM systems include MonoSLAM [13], ORB-SLAM [38], and Parallel Tracking and Mapping (PTAM) [24]. Some semi-dense approaches include Dence Tracking and Mapping (DTAM) [42], SVO [18] and Large Scale Direct (LSD)-SLAM [16].

MonoSLAM is one of the early works demonstrating a real-time monocular SLAM system running at 30Hz. It uses the standard full covariance EKF approach. Representing the map as a single multivariate Gaussian distribution over a region. To initialize the map some prior is given in the form of visual target. Features are initialized once the depth estimate of the landmark coverages over a few frames. The Shi and Tomas corner detector is used to detect features [54]. They are stored as an oriented textures patch and re-projected to into new image frames for matching. This approach is very costly due to the extreme large covariance updates which grow quadratically with the number of landmarks, limiting it to a small region of operation.

PTAM introduced the idea of running feature tracking and landmark mapping as separate

processes using keyframes to speed up operation on low-power hardware. It was developed for an augmented reality application but is equally applicable to a robotic application. This approach allows up to thousands of landmarks to be optimized using windowed bundle adjustment. They used the FAST corner detector along with normalized cross-correlation or Single Shot Detector (SSD) for feature correspondence. The initialization process uses the five-point algorithm with user input. The tracking thread scales well with increased map size, however, the mapping thread is limited by the number of landmarks, again making it only useful in small indoor areas.

ORB-SLAM is a modern SLAM system which uses Oriented-Rotated BRIEF (ORB) for tracking, mapping, re-localization as well as loop closure. It brings together the ideas from PTAM, monocular loop-closure work from [58] and scalability ideas from [57] to achieve a feature complete large-scale SLAM system. Visual Bag of Words (BoW) is used for efficient loop-closure identification and re-localizations.

DTAM is one of the first works which moves away from feature-based SLAM and opts for a dense approach by optimizing all the pixel information instead of just selected patches from the camera sensor. This approach is much more computationally expensive but offers improvements in the presence motion blur, and visually self-similar environments. It is also much more useful in an Augmented Reality (AR) and Virtual Reality (VR) context as the maps are much richer containing the full depth map at every keyframe as well as the texture information. At the same, time this approach is limited by the camera and reflectance models used. The photometric constancy is only valid over an extremely short baseline compared to feature-based approaches, resulting in poor uncertainty in the depth estimates. Dense methods are also more affected by rolling shutter, auto exposure, and lens flares.

LSD-SLAM is similar to DTAM but it makes large-scale application possible by simply performing a generic pose graph optimization using g2o [27] instead of using all the intensity information to incrementally update the map such as with DTAM. The approach comprises of three main parts: tracking the 6-Dof pose changes of the camera, estimating the depth map

by filtering over many small baseline image pairs and map optimization in 7-Dof space to naturally incorporate scale drift for loop-closures.

Finally, SVO uses a combination of dense and feature-based methods to leverage the benefits of both approaches. Feature extraction is only done when a new keyframe is created to initialize the 3D landmark estimates instead of on every single frame. Then the optimization is carried out over these patches, resulting in speed improvements over full dense methods. This was tested on a Micro Aerial Vehicle (MAV) in a primarily downward facing configuration.

There are several reasons for continuing to use feature-based methods for VT&R. This includes lower storage requirements, faster processing time and more invariance over lighting and scene changes. These are all important in the development of a long-term online large-scale visual route following system.

A monocular VT&R system has already been investigated which uses a flat-ground assumption and a known transform between the camera and the ground plane [11]. While the planar assumption of [11] is largely valid in an indoor setting, we provide a solution that makes no assumptions about the nature of the scene, allowing it to function in arbitrary environments. Monocular VT&R has also been implemented on aerial vehicles, including the AR Drone [48], again with an assumed ground plane. It has been deployed on a system that includes no such constraints on a fixed-wing vehicle [66] but under GPS control. This is the first demonstration of a monocular VT&R system that 1) makes no environmental assumptions and 2) includes in-the-loop robot control.

2.2 Long-Term Vision-Based Localization

In the area of long-term visual navigation, a fundamental problem is localizing over time in the presence of natural scene changes as a result of illumination, seasonal, and weather variations. Light Detection And Ranging (LiDAR) systems overcome this limitation, but they are still relatively expensive and require large payload capacities not practical on mass-restricted

systems. Radar also shows significant promise in this regard [7] but are still more expensive than vision-based systems. Cheaper [7] options do not provide as much information as vision or LiDAR and are more susceptible to noise.

Localization solutions can be grouped based on how precise the desired result is: metric localization and topological localization. The approach of interest is the former, which produces an estimated state and uncertainty relative to some internal representation of the environment for the purpose of visual route-following. This is necessary because in order to follow the path closely, the path tracking controller requires a state estimate of the current vehicle location with respect to a reference frame.

In a feature-based framework, an estimated pose can be obtained by solving the point-cloud alignment problem (3D-3D) or Perspective-n-Point (2D-3D) problem. Other approaches use higher dimensional constructs such as lines [47], and objects (SLAM++) [53] for localization in the same manner. Dense methods can minimize the re-projection error using a specified cost function to obtain a relative pose change as well. Topological localization, on the other hand, is primarily used for loop-closure or place recognition tasks. Work in this area is significantly more robust to visual changes than in the area of metric localization.

2.2.1 Topological Localization

Most methods of topological localization use the BoW model in a computer vision context. For each location feature vectors are clustered, these can be considered the words much like words in a dictionary. The number of features in each cluster is similarly analogous to the number of times each word appears in a document. The motivation is that locations with similar appearance should cluster together in feature space. By using such a model, matching different locations simply means comparing the feature vectors or some type of aggregated representation of these feature vectors such as a histogram.

Cummins *et al.* proposed Fast Appearance Based Mapping (FabMap) [12] using a generative BoW in a probabilistic framework. The system is able to capture the relationships between

combinations of appearance occur together. It shows impressive performance even when there are few features in common between two locations and is able to reject localizations which have many features in common due to visual aliasing. The efficiency of the proposed system makes it an ideal loop closure detector. However, performance suffers in the presence of lighting changes.

Milford *et al.* [37] proposed Sequence SLAM (SeqSLAM) which use a sequence of images for localization. By constructing a similarity matrix and finding the sequence with the best score, the system is able to correctly localize in the presence of significant appearance changes. While effective, these topological systems are not able to provide the precision required for vision-in-the-loop navigation.

McManus *et al.* [36] presented the idea of scene signatures for localization. The scene signatures are patches in the image which can be corresponded to each other using an SVM classifier. They termed this approach a weak localizer due to the fact that it is not possible to precisely compute the transformation between the images directly on these patches. A 3D position can still be associated with each patch in order to produce an approximate estimate. This proves to be quite effective at localization across lighting and seasonal changes, but it makes a trade-off on the accuracy of the localization.

2.2.2 Metric Localization

Vision-based metric localization can be achieved by matching point features in images taken at different times and computing the relative pose change of the camera. To obtain these point features, a detection scheme is used to find the most salient points in the environment. These points should ideally correspond to the same triangulated landmarks in the environment irrespective of illumination or viewpoint changes. The information around these points can be summarized with a description function and then matched using a distance function. The inlier matches can then be used to estimate the six-degree-of-freedom (6DoF) transformation between the two camera positions.

2.2.3 Better Descriptors

The main issues with feature-based methods is data correspondence across visual changes in the scene. Valgren *et al.* [65] examined the use of SIFT and SURF descriptors for long-term navigation. They conclude that U-SURF resulted in the best performance, but ultimately using local feature matching alone is not sufficient for cross-seasonal metric localization.

It is difficult to deal with seasonal changes, but illumination changes and shadows are easier to deal with. Techniques such as illumination-invariant images [35] and colour-constant images [34] result in more stable but often a smaller number of feature matches. Other image processing techniques such as contrast limited adaptive histogram equalization (CLAHE) [67] creates an order of magnitude more matches by bringing out more details in the image.

Binary descriptors such as BRIEF [5], ORB [52], and BRISK [29] are computed by comparing the intensity values at various positions within a patch around the image feature. The number of possible positions for such a computation can be substantial, especially for larger image patches. The authors of BRIEF drew positions from common distributions and chose the best ones. The authors of ORB chose comparisons with high variance. BRISK uses a pattern that is composed of concentric rings. The sampling strategy has a significant effect on the result of matching, and so GRIEF [26] was devised to find the best positions within a patch given pre-labeled data. This employs an evolutionary algorithm which seeks to maximize the number of comparisons which result in true positive matches.

Different techniques have been applied for learning better visual descriptors to improve their performance. Two examples are: convex optimization [56] and convolution neural networks (CNN) [6], [55]. Floating-float descriptors such as Learned Invariant Feature Transform (LIFT) have also been learned using Siamese Networks [68]. These learned descriptors are quite robust to lighting, viewpoint, deformations, and small seasonal changes. The only drawback is that they are much more computationally intensive than binary descriptors. The training process is also significantly more expensive compared to the evolutionary algorithm used in GRIEF.

2.2.4 Map Management

Dayoub *et al.* [14] proposed a system that employs the idea of short and long-term memory to forget old features and add new ones. ORB-SLAM prunes the map periodically to remove feature using a survival of the fittest strategy. Churchill *et al.* [9] introduces the notion of saving multiple experiences in problematic areas and localizing to them all in parallel. Given various experiences then comes the question of which of the experiences to use and prioritize, this is explored by Linegar *et al.* in [30]. Multi-experience VT&R [45] similarly uses bridging experiences to overcome the presence of natural scene changes. Every time a vehicle drives through the environment, a completely new map is generated with respect to the original or privileged experience. This allows the system to match to any of the stored experiences. It has been demonstrated to work across seasons from fall to winter and into the springtime [46].

2.2.5 Image Transformations

Neubert *et al.* [41] proposed a method of predicting the appearance change and matching the predicted image against the live images. This uses vocabularies of super pixel (SP) or SP-apearance change prediction (ACP) to map words from one environmental condition to a different condition. Combined with SeqSLAM or other topological localization systems it can significantly improve the performance even across seasons.

More recently, to deal with the problem of cross-seasonal localization the approach of image to image transformations have also been explored in [23] then later applied to a mapping and localization task in [10] using conditional adversarial networks. Similarly, a cycle consistent adversarial networks is used in [69] and later [49] for cross-seasonal localizations to great effect. These techniques rely on learning a mapping from one type of appearance to another using a neural network. The generalizability of the approach still need to be validated.

The work presented in this thesis differs from the above methods by tailoring the description function to the environment. We hypothesize that adapting the description function to the environment leads to improved localization performance. This is similar in principle to us-

ing individual SVMs for each landmark, which proves to be quite robust [31]. However, we demonstrate this within a traditional feature-based navigation system with binary descriptors for improved runtime performance.

Chapter 3

Stereo Visual Teach & Repeat

3.1 System Overview

VT&R is a keyframe-based mapping and localization framework for visual route following [19]. A sliding-window filter approach is employed to ensure accurate pose estimation while maintaining real-time performance. There are two main states the vehicle can be in: teach and repeat. The teach phase is essentially map creation using Visual Odometry (VO) and windowed bundle adjustment. This is usually achieved by leveraging a human operator for the initial traversal but it can also be accomplished by using GPS or an external exploration mechanism. In the repeat phase the vehicle is able to metrically localize to the map and navigate both

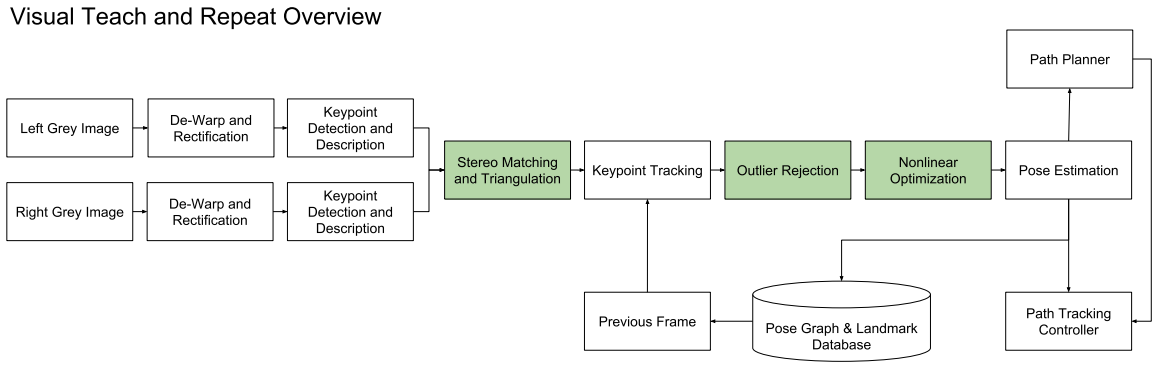

Figure 3.1: Overview of Visual Teach and Repeat System

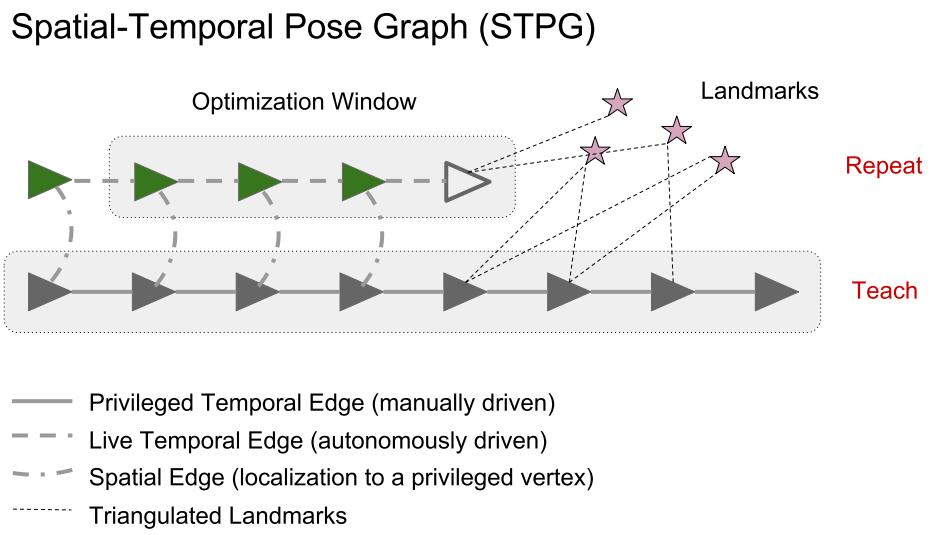

Figure 3.2: Spatial Temporal Pose Graph (STPG) used to store all relevant data to VT&R.

forwards and backwards along the taught path. Building upon this, a network of paths can be created by branching off existing paths. The vehicle is then able to autonomously plan and navigate to any location within the network of paths using a planner and path tracking controller as shown in Figure 3.1.

The underlying data structure used by the system is a Spatio-Temporal Pose Graph (STPG) (see Figure 3.2). The vertices in the graph correspond to a specific pose at a particular instant in time. All the relevant sensor data are stored in the vertex. For a vision system, this includes the camera images, 2D feature positions, and 3D landmark positions. The edges in the graph contain the estimated transformations and uncertainties between vertices obtained by solving an optimization problem. The parts of the graph constructed from the teach phase are considered "privileged" because this is the path known to be safe and obstacle free. The edges from a single repeat are called temporal edges. The edges linking different repeats are called spatial edges. Temporal edges are generated from VO whereas spatial edges are generated from localization.

At a basic level, the system is built from two main pipelines: odometry and localization. Terrain assessment is also an important aspect of the system but it is not necessary for basic

operation. Its main purpose is to identify objects which were not present during the original teach phase and safely stop. The two pipelines can be subsequently broken down into modules explained below. This is the system used for the development of the place-dependent descriptor as described in Chapter 5.

3.2 Mathematical Background

Robots move in three-dimensional space, therefore we need an appropriate parameterization for representing the state of the robot at any given time. The position of the robot can simply be expressed as a vector. For orientation there are many choices, some common representations include quaternions, Euler angles, and rotation matrices. Here we choose to use the rotation matrix formulation due to the fact pose changes can simply be expressed as a matrix multiplication.

Rotation matrices are part of the special Euclidean group, $SO(3)$, and is defined as the following, with two constraints to ensure it is a proper rotation (Eq. 3.1).

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3\times 3} \mid \mathbf{C}\mathbf{C}^T = 1,\, \det \mathbf{C} = 1 \right\}. \tag{3.1}$$

If we combine the position vector with a rotation matrix we obtain the special orthogonal group, $SE(3)$, which we will refer to as a pose (Eq. 3.2). Similarly, we constrain the elements so that it is a proper transformation matrix.

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} \in \mathbb{R}^{4\times 4} \mid \mathbf{C} \in SO(3),\, \mathbf{r} \in \mathbb{R}^3 \right\}. \tag{3.2}$$

Both the set of rotation matrices and poses as defined previously are matrix Lie groups. In order to use the pose representation in an optimization problem we use the Lie algebra associated with each Lie group, $\mathfrak{so}(3)$ (Eq. 3.5) and $\mathfrak{se}(3)$ (Eq. 3.3). By using these two representations we overcome the issues of overparameterization as well as discontinuities during

the optimization process.

$$\mathfrak{so}(3) = \left\{ \boldsymbol{\Phi} = \boldsymbol{\phi}^{\wedge} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\phi} \in \mathbb{R}^{3} \right\}. \tag{3.3}$$

where

$$\boldsymbol{\phi}^{\wedge} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \tag{3.4}$$

$$\mathfrak{se}(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^{\wedge} \in \mathbb{R}^{4\times 4} \mid \boldsymbol{\xi} \in \mathbb{R}^{4} \right\}. \tag{3.5}$$

where

$$\boldsymbol{\xi}^{\wedge} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^{\wedge} = \begin{bmatrix} \boldsymbol{\phi}^{\wedge} & \boldsymbol{\rho} \\ \mathbf{0}^{T} & 0 \end{bmatrix} \tag{3.6}$$

To convert between the matrix Lie group and Lie algebra we can use the exponential and logarithmic map operators. It should be noted that multiple elements in $\mathfrak{so}(3)$ maps to the same element in $SO(3)$ and similarly for $\mathfrak{se}(3)$ and $SE(3)$. In other words, going from the Lie algebra to the Lie group is a surjective-only mapping. For more details and derivations refer to [3].

$$\mathbf{C} = \exp(\boldsymbol{\phi}^{\wedge}), \boldsymbol{\phi} = \ln(\mathbf{C})^{\vee} \tag{3.7}$$

$$\mathbf{T} = \exp(\boldsymbol{\xi}^{\wedge}), \boldsymbol{\xi} = \ln(\mathbf{T})^{\vee} \tag{3.8}$$

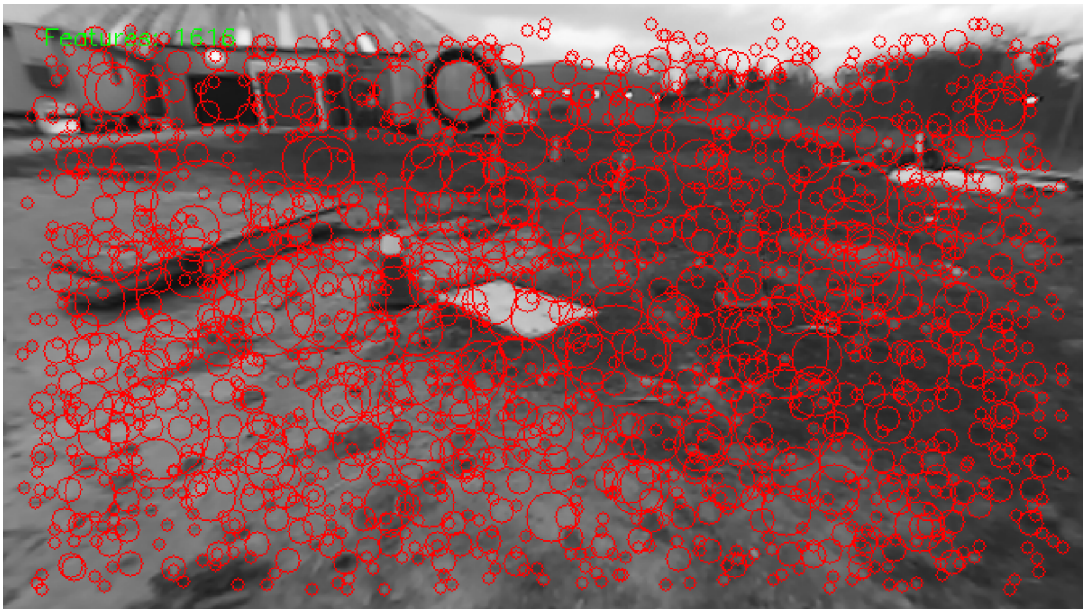

Figure 3.3: Feature detection on a natural scene using SURF.

3.3 Odometry Pipeline

The odometry pipeline estimates the motion of the camera and vehicle through the environment by tracking visual features. With the two sets of 3D landmarks locations associated with the visual features, we need to find the optimal transformation between the landmarks. The closed form solution to this least squares problem is presented in [21] by Horn *et al.*. To reject outliers we solve this problem aided by RANdom SAmple Consensus (RANSAC). We will refer to this problem as the point-cloud alignment problem from here on. Using this as an initial solution, the landmark and vehicle poses are then further refined by solving the bundle adjustment problem.

3.3.1 Feature Extraction

This stage includes image pre-processing, feature detection and feature description. The input is the raw stereo images and the output is a list of sparse features positions for each image and descriptors that summarize the pixel information at those particular locations. The usual pre-processing performed is the conversion of the image to a single-channel greyscale repre-

sentation. Other types of pre-processing includes: colour constancy conversion and histogram equalization. These steps reduce the effects of lighting on the images to improve feature correspondences.

After the image conversion feature extraction is performed. This can be broken down into two parts: detection and description. Feature detectors can be classified into corner detectors (Harris, FAST, Shi and Tomasi), edge detectors (Canny, Sobel, Prewitt) and blob detectors (SURF, Laplacian of Gaussian, Maximally Stable Extremal Regions). Features description schemes include: Speeded-Up Robust Features (SURF), SIFT, Binary Robust Independent Elementary Features (BRIEF), and ORB to name a few. The two algorithms currently used in VT&R are SURF and ORB. A visual representation of this is presented in Figure 3.3 using the SURF detector. Each red circle denotes a point of interest or keypoint. The size of each feature is represented by the radius of the circle. The feature descriptor for SURF usually a 64-dimensional feature vector.

3.3.2 Landmark Triangulation

After detecting keypoints of interest, we must solve for the 3D coordinates of the landmarks associated with each keypoint. Assuming the stereo camera has the same intrinsic for both cameras and a baseline distance of b, the equation for projecting any 3D landmarks (left camera center as origin) onto the two image planes is given by:

$$\begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ f_u & 0 & c_u & -bf_u \\ 0 & f_v & c_v & 0 \end{bmatrix} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{3.9}$$

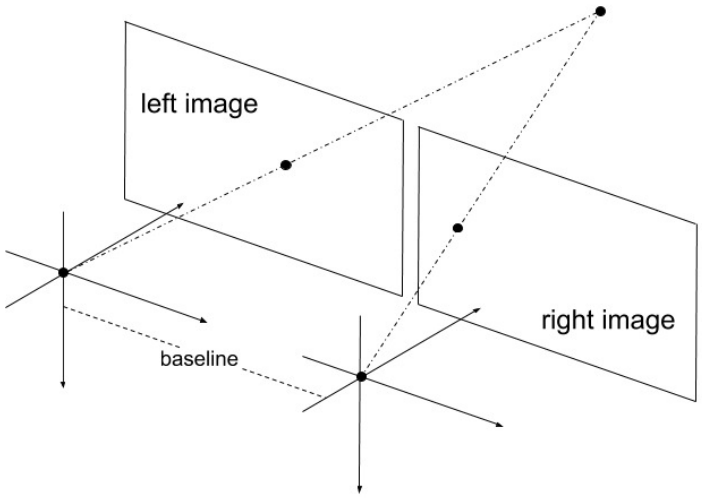

Figure 3.4: Stereo camera model

3.3.3 Feature Matching

The feature matching process exhaustively matches the descriptors from the feature extraction stage back to the previous keyframe. Cosine distance is used for floating-point descriptors such as SURF and Hamming distance is used for binary descriptors such as ORB. Some heuristics can be applied to speed up the matching process. We can limit the search space by using a constant velocity assumption and only searching in the region where the feature is expected to be present. Limits on the descriptor threshold and analysis of metadata associated with the keypoint can also be used to speed up the matching. The result is a list of correspondences based on the appearance of the patch around the keypoint locations in both images as shown in Figure 3.5.

3.3.4 Point-Cloud Alignment Problem

With the initial set of correspondences, RANSAC is used to reject any outliers and provide an initial estimate of the pose change from the previous keyframe. The point-cloud alignment problem can be solved using a rotation matrix formulation in closed form using only three points assuming consistent scale [21]. The three points also must not be collinear as this is a degenerate case.

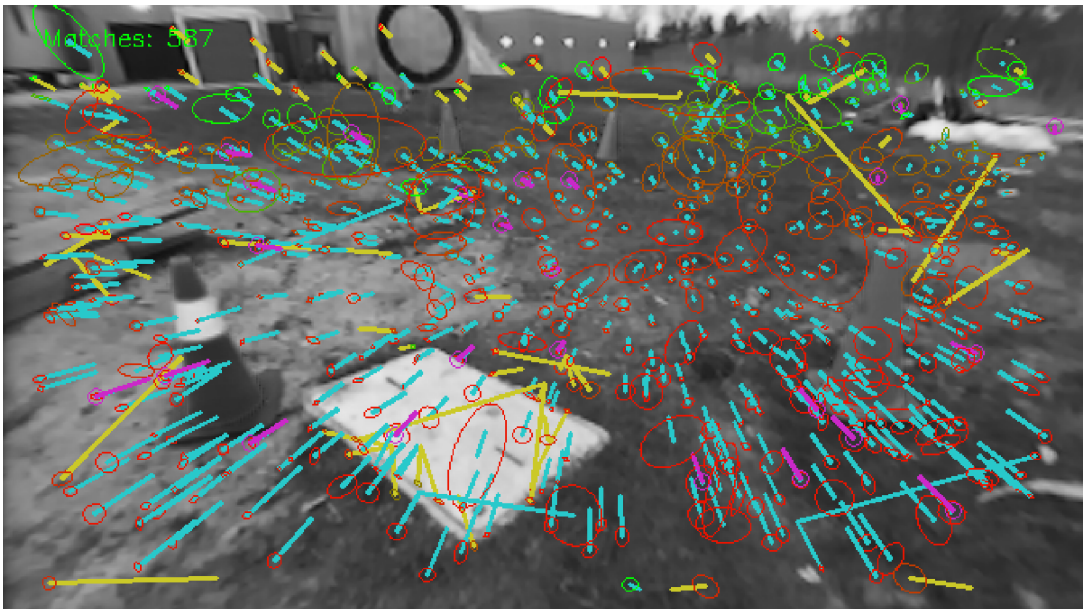

Figure 3.5: Raw feature matches from frame to frame. The light-blue lines represent the inlier set and the yellow lines represent the outlier set. The size of the circles represent the uncertainty associated with each landmark.

The problem formulation is: given two reference frames, $\mathcal{F}_{\rightarrow a}$ and $\mathcal{F}_{\rightarrow b}$, both observing the same set of landmarks, find the transformation (rotation and translation) between the two reference frames. In other words, given two set of measurements of the same landmarks, l_j, from two frames $\{r_a^{l_j a}, r_b^{l_j b}\}$, find $\mathbf{T_{ba}}$.

3.3.5 Keyframe Optimization

We can then further refine the estimated transformation, $\mathbf{T}$, by using an iterative approach by minimizing a cost function based on the re-projective errors of the landmarks using the transformation matrix formulation presented previously:

$$J(\mathbf{T}) = \frac{1}{2}\sum_{j=1}^{N} w_j(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j)^T(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j) \tag{3.10}$$

where w_j is the positive scalar weighting on the landmark. The vector $\boldsymbol{y}_j$ is the coordinate of the feature in the image frame and the vector $\boldsymbol{p}_j$ is the coordinate of the landmark location, both in homogeneous coordinates. This optimization problem can then be solved iteratively using

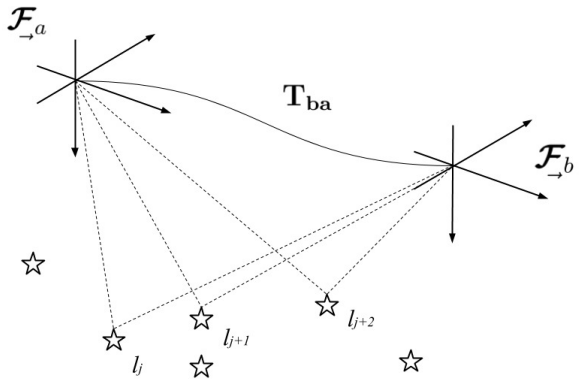

Figure 3.6: Point-Cloud Alignment Problem

a method such as Gauss-Newton. We apply a small perturbation, $\boldsymbol{\epsilon}$, to the current operating point, $\mathbf{T}_{\text{op}}$, and solve for the optimal perturbation which reduces the cost function. This can be done by finding the derivative of the cost function with respect to the perturbation.

$$\mathbf{T} = \exp(\boldsymbol{\epsilon}^{\wedge})\mathbf{T}_{\text{op}} \approx (\mathbf{1} + \boldsymbol{\epsilon})\mathbf{T}_{\text{op}} \tag{3.11}$$

The optimal update can be solved in closed form:

$$\boldsymbol{\epsilon}^{\star} = \boldsymbol{\mathcal{T}}_{\text{op}}\boldsymbol{\mathcal{M}}^{-1}\boldsymbol{\mathcal{T}}_{\text{op}}^{T}\mathbf{a} \tag{3.12}$$

where $\boldsymbol{\mathcal{T}}$ is the adjoint of $\mathbf{T}$ as defined in [3]. The terms $\mathbf{p}_j$ and $\mathbf{w}_j$ are the non-homogenous vector representations for $\boldsymbol{p}_j$ and $\boldsymbol{w}_j$, respectively. These equations are taken from the book State Estimation for Robotics [3],

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\mathbf{p}^{\wedge} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{p}^{\wedge} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}, \ \mathbf{a} = \begin{bmatrix} \mathbf{y} - \mathbf{C}_{\text{op}}(\mathbf{p} - \boldsymbol{r}_{\text{op}}) \\ \mathbf{b} - \mathbf{y}^{\wedge}(\mathbf{p} - \boldsymbol{r}_{\text{op}}) \end{bmatrix} \tag{3.13}$$

where:

$$w = \sum_{j=1}^{N} w_j,\ \mathbf{p} = \frac{1}{w}\sum_{j=1}^{N} w_j \mathbf{p}_j,\ \mathbf{I} = \frac{-1}{w}\sum_{j=1}^{N} w_j (\mathbf{p}_j - \mathbf{p})^{\wedge}(\mathbf{p}_j - \mathbf{p})^{\wedge} \tag{3.14}$$

$$\mathbf{b} = \left[\mathrm{tr}(\mathbf{1}_i^{\wedge}\mathbf{C}_{\mathrm{op}}\mathbf{W}^T)\right]_i,\ \mathbf{y} = \frac{1}{w}\sum_{j=1}^{N} w_j \mathbf{y}_j,\ \mathbf{W} = \frac{-1}{w}\sum_{j=1}^{N} w_j (\mathbf{y}_j - \mathbf{y})(\mathbf{p}_j - \mathbf{p})^T \tag{3.15}$$

3.3.6 Windowed Optimization

In addition to refining the pose estimate of the vehicle, we can refine the estimates for the landmark positions as well as the vehicle pose by constructing an optimization problem over a set of keyframes (vertices). This is known as the bundle adjustment problem, with the incorporation of a motion model this becomes the classic SLAM problem.

$$J(\mathbf{T}, \mathbf{p}) = \frac{1}{2}\sum_{jk} (\mathbf{y}_{jk} - \mathbf{g}_{jk}(\mathbf{T}, \mathbf{p}))^T \mathbf{R}_{jk}^{-1} (\mathbf{y}_{jk} - \mathbf{g}_{jk}(\mathbf{T}, \mathbf{p})) \tag{3.16}$$

Similar to Equation (3.10), the cost function is constructed as the sum of squared error terms using the observation model $\mathbf{g}_{jk}(\cdot)$, meaning the projection of landmark k into the camera frame at pose j. Any landmarks not observed can have an error term set to zero. The matrix $\mathbf{R}_{jk}$ is the covariance associated with each measurement.

3.3.7 Vertex Creation

A new vertex is created based on two primary criteria: pose change and number of tracked features. Pose change refers to both translational and rotational movements. Visual descriptors are only invariant under small pose changes, when the viewing angle is too large it becomes difficult to obtain good feature correspondences from frame to frame. The number of tracked features is a direct measure of this and is similarly used in case the correspondence count becomes too low for proper pose estimation.

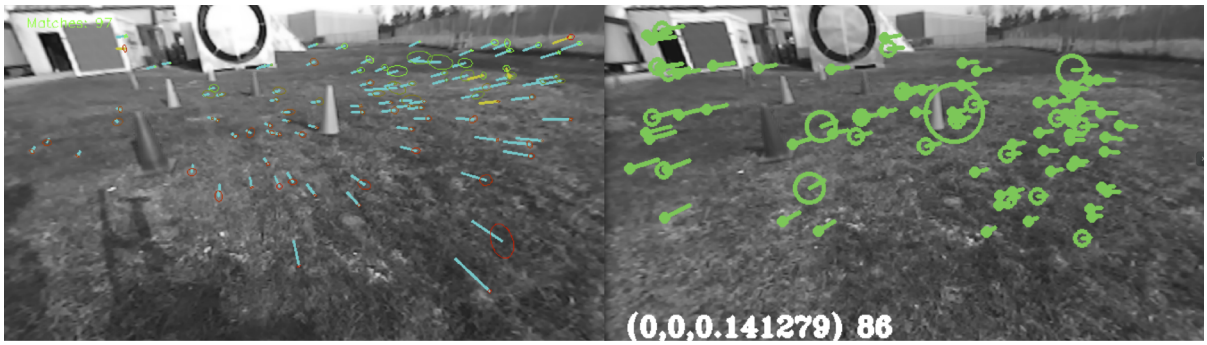

Figure 3.7: Visual Odometry and Localization post-RANAC matches

3.4 Localization Pipeline

The localization pipeline is very similar to the visual odometry pipeline. The difference is that instead of estimating pose with respect to the previous keyframe, we are estimating pose relative to a keyframe generated in a previous traversal (map). The first step is landmark migration which transforms all the landmarks in the map to the same reference frame. Then feature matching and the point-cloud alignment problem is solved in the exact same manner as before. The final posterior estimate comes from an optimization problem similar to Equation 3.16 but with the incorporation of a prior error term from VO.

Formally, we seek the posterior transform and uncertainty $\{\hat{\mathbf{T}}_{ab}, \hat{\mathbf{\Sigma}}_{ab}\}$ from the closest vertex V_b in the map to the current live view V_a. This can be computed by minimizing the sum of the squared re-projective error of the landmarks $\mathbf{e}_i$ and the difference between the prior and the true state $\mathbf{e}$. The logarithmic map and $\vee$ operator are as defined in [3].

$$J(\mathbf{T}_{ab}) = \frac{1}{2}\sum_{i=1}^{N} \mathbf{e}_i^T \mathbf{R}_i^{-1} \mathbf{e}_i + \frac{1}{2}\mathbf{e}^T \mathbf{R}^{-1} \mathbf{e}, \tag{3.17}$$

$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{g}(\mathbf{T}_{ab}\boldsymbol{p}_{b,i}), \mathbf{e} = \ln(\check{\mathbf{T}}_{ab}\mathbf{T}_{ab}^{-1})^{\vee}. \tag{3.18}$$

Chapter 4

Monocular Visual Teach & Repeat

4.1 Overview

Monocular VT&R is built using the same framework as stereo VT&R described in Chapter 3. The main differences are highlighted using the green blocks in Figure 4.1. The key difference results from the fact that landmark triangulation is no longer directly possible from the pair of live images, instead they must be computed using the motion of the camera through space. An initialization procedure is also required to obtain an initial set of 3D landmark positions. Each of the blocks which differ is explained below.

4.1.1 Monocular Initialization

To initialize the VO, SURF keypoints and descriptors [4] are extracted from the first image and then matched against those from subsequent frames. Both an Essential matrix (Eq. 4.1) and 2D Homography matrix (Eq. 4.2) are computed using Maximum Likelihood Estimation SAmple Consensus (MLESAC) [62] to estimate a relative transformation from the first frame to the current live view.

Given two images viewing the same scene, if $\mathbf{x}'$ and $\mathbf{x}$ are the coordinates of the features corresponding to the same landmarks in homogeneous coordinates we can compute either a

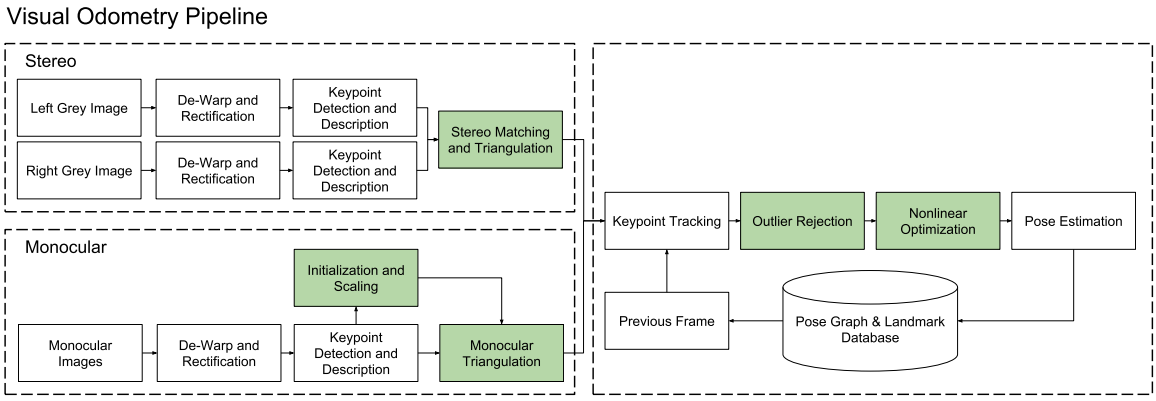

Figure 4.1: Comparison of the stereo visual odometry pipeline compared to the monocular visual odometry pipeline. The differences between the two are highlighted in green.

homography matrix $\mathbf{H}$ or an essential matrix $\mathbf{E}$. The homography matrix is only valid when the scene is flat, whereas the Essential matrix is more general but degenerate when the scene is flat. Therefore depending on the environment, we must choose one over the other.

$$\mathbf{x'}^T\mathbf{E}\mathbf{x} = 0, \tag{4.1}$$

$$\mathbf{x} = \mathbf{H}\mathbf{x}'. \tag{4.2}$$

Both solutions are examined and the Geometric Robust Information Criterion (GRIC) test [63] selects the optimal solution. It should be noted that for initialization to occur the user must command the vehicle manually by driving in a straight line as it is not handled automatically by the controller due to safety issues.

Once the inlier count for each frame-to-frame matching drops below a threshold (an analog for translational and angular motion), landmarks are triangulated and the pair of frames are placed as the first two vertices in the graph, with the computed transformation inserted in the edge. To initialize the scale, wheel odometry is used to estimate the translational distance between the two vertices, which is then used to find the appropriate scaling parameter. This scaling factor is then applied to the transformation between the first two vertices and all

landmarks positions. This magnitude information can come from any source not only wheel odometry, examples include integrated IMU measurements, known distance from the scene to the camera, or GPS measurements.

4.1.2 Monocular Visual Odometry

After initialization is completed, new features are extracted from the live images and matched to the last keyframe in the graph. These features are triangulated using an estimated transform from VO. The VO update is generated by solving the Perspective-Three-Point (PnP) problem aided with RANSAC for outlier rejection. A solution is presented in [28] by Lepetit *et al.*.

Given four keypoints, $\mathbf{p'}_i$, and their corresponding 3D position, $\mathbf{P}_i$, the solution can be solved to scale so external measurements for scaling are no longer required. The scale does start to drift over longer distances, which can be problematic in certain scenarios.

To decrease the search space for feature correspondences, a trajectory estimate using a constant-velocity assumption is used for feature matching. As new keyframes are created, a windowed bundle adjustment optimization using the Simultaneous Trajectory Estimation And Mapping (STEAM) library [1] is performed over a variable number of the last few vertices. This refines the estimates for the landmark and camera poses. The scale is held constant during the optimization based on the distance the vehicle has traveled over that window.

4.1.3 Perspective-n-Point Problem

Unlike in the case with stereo where we solved the point-cloud alignment problem with the aid of RANSAC using 3D-3D correspondence. In the monocular scenario, a 3D-2D correspondence is used to solve for a relative pose. This is mainly due to the large uncertainties associated with the actual landmarks positions in the monocular case.

The problem formulation is: given two reference frames, $\mathcal{F}_{\vec{}a}$ and $\mathcal{F}_{\vec{}b}$, both observing the same set of landmarks, find the transformation (rotation and translation) between the two reference frames. In other words,given a set of measurements of the same landmarks, l_j and their

projections, p_j from two frames $\{r_a^{l_j a}, r_b^{l_j b}\}$ respectively, find $\mathbf{T}_{ba}$.

In the case of an indoor environment such as a warehouse, the problem can be further constrained to a 3-DoF solution because the vehicle is only expected to move along a 2D plane. We present the analytical solution for the 2D case. The minimum number of correspondences required is 2 which over constrains the 3-DoF that is present in the planar case. Each correspondence results in 2 equation for a total of four equations. We can stack them to obtain the following system of equations (Eq. 4.4).

$$\begin{bmatrix} \mathbf{p'}_1 & \mathbf{p'}_2 \end{bmatrix} = \mathbf{KT} \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} u_1 & u_2 \\ v_1 & v_2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & t_1 \\ \sin(\theta) & \cos(\theta) & 0 & t_2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ z_1 & z_2 \\ 1 & 1 \end{bmatrix} \tag{4.4}$$

To solve for the transformation $\mathbf{T}$ we can gather the unknown terms $\sin(\theta), \cos(\theta), t_1, t_2$ and rearrange to get the following system which can be solved using any least squares method.

$$\mathbf{Aw} = \mathbf{b}, \tag{4.5}$$

$$\begin{bmatrix} -x_1 f_y & -y_1 f_y & 0 & -f_y \\ -y_1 f_x & x_1 f_x & f_x & 0 \\ -x_2 f_y & -y_2 f_y & 0 & -f_y \\ -y_1 f_x & x_1 f_x & f_x & 0 \end{bmatrix} \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} z_1(c_y - y_1) \\ -z_1(c_x - x_1) \\ z_2(c_y - y_2) \\ -z_2(c_x - x_2) \end{bmatrix} \tag{4.6}$$

Alternatively since $\sin(\theta), \cos(\theta)$ are dependent on θ based on the identity $\sin(\theta)^2 + \cos(\theta)^2 = 1$, we can obtain the following system reperameterized in terms of elements of $\mathbf{A}$ and $\mathbf{b}$.

$$\mathbf{Cx} = \mathbf{d}, \tag{4.7}$$

$$\mathbf{C} = \begin{bmatrix} M & 0 & A_{0,3} \\ (A_{1,1}A_{2,0} - A_{1,0}A_{2,1})M & A_{0,1}A_{1,2}A_{2,0} - A_{0,0}A_{1,2}A_{2,1} & A_{0,0}A_{1,1}A_{2,3} - A_{0,1}A_{1,0}A_{2,3} \\ (A_{1,1}A_{3,0} - A_{1,0}A_{3,1})M & N & 0 \\ (A_{2,1}A3,0 - A_{2,0}A_{3,1})M & A_{0,0}A_{2,1}A_{3,2} - A_{0,1}A_{2,0}A_{3,2} & A_{0,1}A_{2,3}A_{3,0} - A_{0,0}A_{2,3}A_{3,1} \end{bmatrix} \tag{4.8}$$

$$\mathbf{x} = \begin{bmatrix} \frac{\theta}{\sqrt{(\theta^2+1)}} & t_1 & t_2 \end{bmatrix}^T, \tag{4.9}$$

$$\mathbf{d} = \begin{bmatrix} b_0 \\ A_{0,0}A_{1,1}b_2 - A_{0,1}A_{1,0}b_2 - A_{0,0}A_{2,1}b_1 + A_{0,1}A_{2,0}b_1 \\ A_{0,0}A_{1,1}b_3 - A_{0,1}A_{1,0}b_3 - A_{0,0}A_{3,1}b_1 + A_{0,1}A_{3,0}b_1 \\ A_{0,0}A_{2,1}b_3 - A_{0,1}A_{2,0}b_3 - A_{0,0}A_{3,1}b_2 + A_{0,1}A_{3,0}b_2 \end{bmatrix}, \tag{4.10}$$

$$M = \sqrt{A_{1,1}A_{0,0} + A_{0,1} * A_{0,1}}, \tag{4.11}$$

$$N = A_{0,0}A_{1,1}A_{3,2} - A_{0,0}A_{1,2}A_{3,1} - A_{0,1}A_{1,0}A_{3,2} + A_{0,1}A_{1,2}A_{3,0} \tag{4.12}$$

The solution for the general 3D case is more involved and details on the derivation can be found from the Lepetit *et al.* paper [28].

4.2 Experimental Setup

The monocular system is tested using the Clearpath Husky platform and the Stereo Labs ZED camera (Figure 4.2). Only the left image is used for the purposes of the experiment. The experiments are carried out in the UTIAS MarsDome, an 1100 square meter dome that simulates the Martian surface. An upward orientation is chosen for the camera to test the ability of the

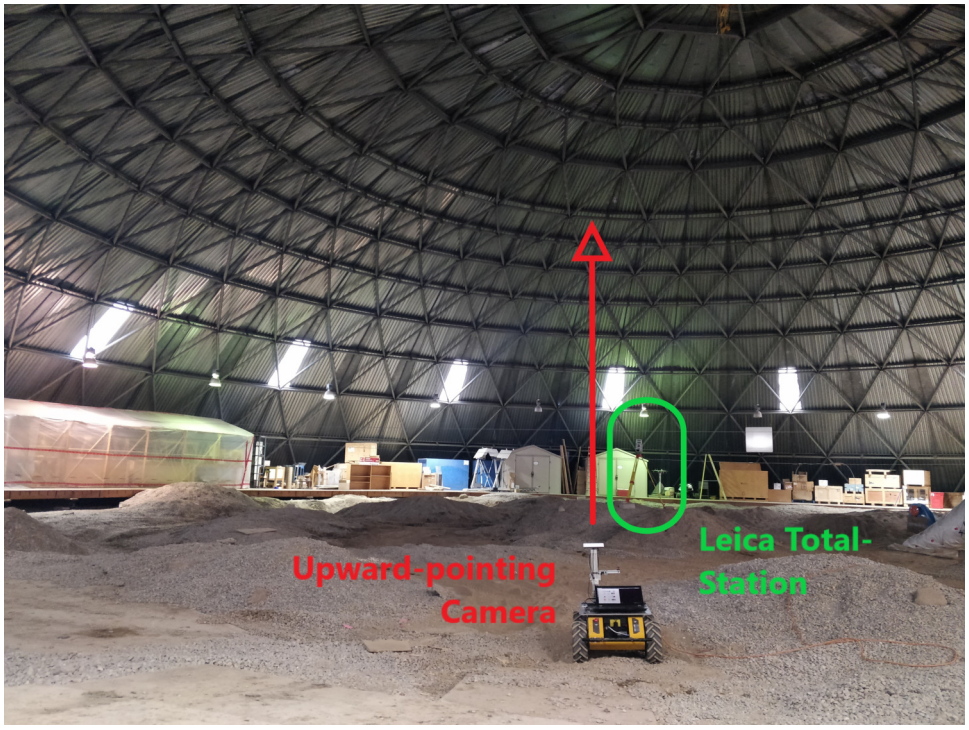

Figure 4.2: Test environment in the MarsDome for the upward monocular VT&R system.

system to deal with landmarks at varying depths. The ceiling is also less likely to change in appearance over time making it a good demonstration for performance in an indoor warehouse environment.

A path 110 meters in length was taught by a human operator and this path was repeated 10 times autonomously. Images along the path are shown in Figure 4.2 and 4.6. The ground is uneven and contains sections of gravel and sand. The path provides a rigorous test with complex scene geometry as well as aggressive turns and poor lighting. This was done to understand the limitations of the approach and examine some failure cases of the system.

We use a basic version of a Model Predictive Control (MPC) controller [44] to minimize the path error between the originally taught path and the current robot pose, with the learning and speed scheduling disabled. As is well known, PnP solutions for monocular cameras suffer from a degeneracy in cases of pure rotation [28]. In particular, this means new landmarks cannot be triangulated on the generation of new keyframes. For the Husky rover (which is a skid-steer vehicle), pure rotations are a common occurrence; especially for a vehicle with an upward

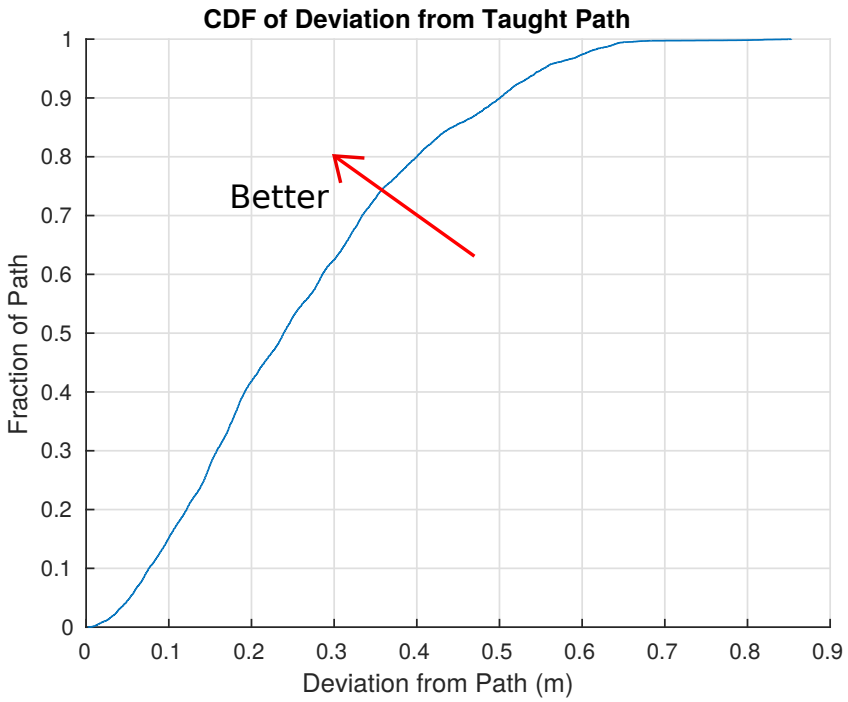

Figure 4.3: The cumulative distribution function of deviations from the taught path. Can be read for as for Y percentage of the path traveled, the deviation is smaller than X meters. For 90% of the 1.1 kilometers, the path tracking error was less than 0.5 meters.

facing camera. We partially mitigate this effect by deliberately placing the camera laterally from the central yaw-axis of the skid-steer mechanism. However, this offset is generally not enough to reduce the effect of pure rotational motions of the vehicle. During the teach phase, the operator is able to avoid these situations by driving the vehicle in an Ackerman-steered style. We also modify the MPC controller so that it does not favour performing on-the-spot turns to reduce the occurrence of near-pure rotational motion.

4.3 Results

The trajectory for all 10 repeats are presented in Figure 4.5. Examining the cumulative error along the path (Figure 4.3), during 90% of the traversal the tracking error was less than 0.5 meters. The cross-track error (lateral deviation) over the entire path is shown in Figure 4.7 with a maximum deviation of 0.85 meters and an average of 0.26 meters.

In general, turns temporarily increase the path tracking error when the curvature of the path exceeds 0.3 m^{-1}. The ceiling is around 15 meters high, this means even with large deviations from the path, the viewpoint did not change drastically. This means the uncertainties of the landmark positions and hence the vehicle position is relatively large, but the mean estimate is close to the desired state. This results in a monotonic increase in path tracking error from the start to around 60 meters into the path despite successful localizations.

After small turns the path tracking error generally decreases as seen around 25 meters, 60 meters and 80 meters. This is because the turns create larger viewpoint changes from frame to frame, allowing the system to localize to a greater degree of certainty. This is demonstrated by the fact path tracking error generally increases right before a turn then decrease after a turn as seen in Figure 4.7. This is also partly due to the fact we limit the max angular rate of the vehicle to avoid pure rotational motions.

There were four manual interventions required during the 10 autonomous repeats as shown in Figure 4.7. All four cases were caused by failed triangulations and the system requiring re-initialization. They are denoted by the red circles around 60 meters and 100 meters into the path. Each one amounts to about a meter of manual driving until re-initialization could occur. This results in a 99.6% autonomy rate over the total distance traveled, excluding the manual initialization required at the start of all repeats.

In addition to high path curvature, another common reason for triangulation failure is uneven terrain. This causes the vehicle and hence the camera to oscillate. Due to the large distance between the camera and the scene, small oscillations result in large feature movements in the camera frame. This is problematic for VO and successful triangulations.

High curvature sections along the path are also often a result of piles of sand or rocks in the way of the vehicle as shown in Figure 4.6. This means even small deviations off the taught path can result in drastic viewpoint changes. All these factors combined resulted in the four triangulation failures at around 60 and 100 meters into the path.

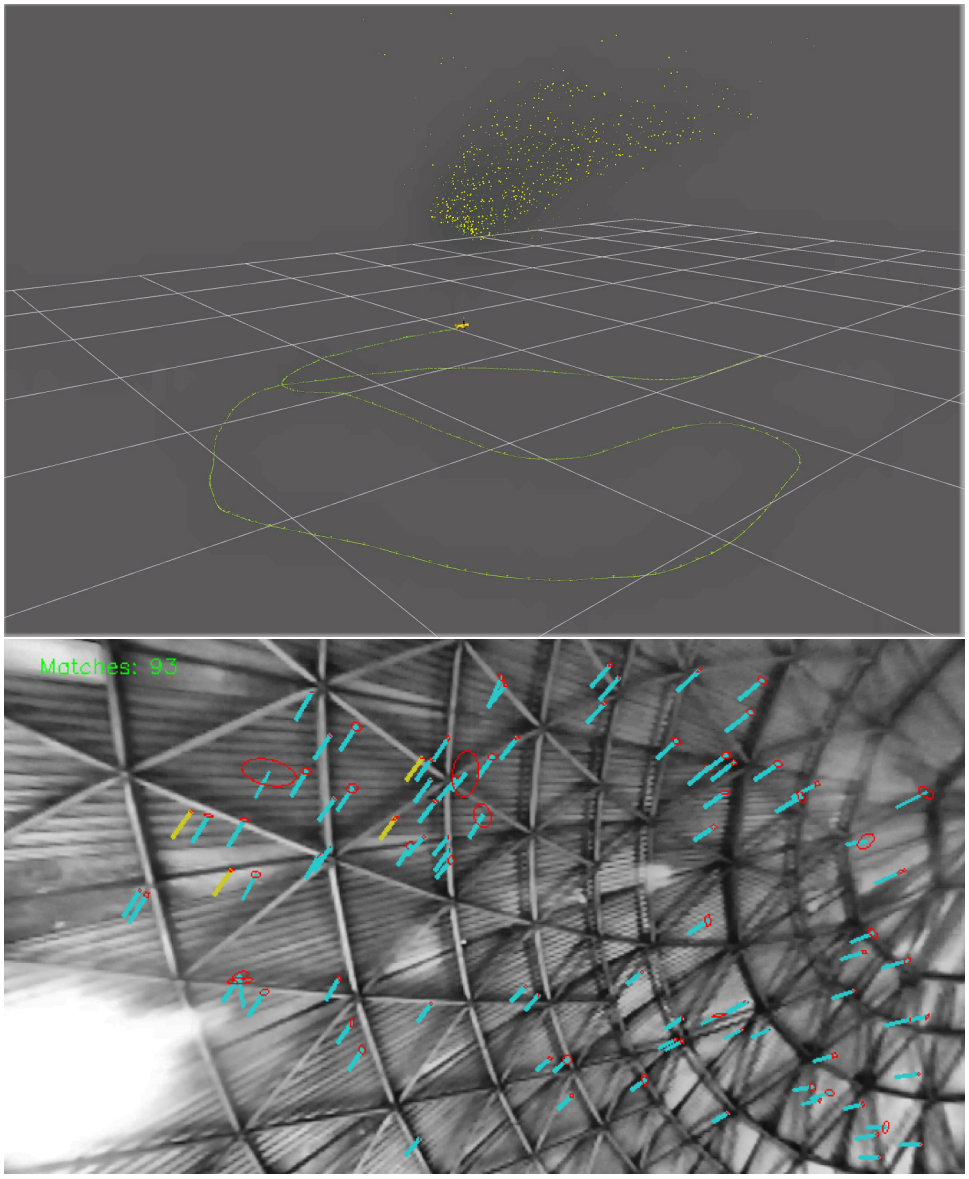

Figure 4.4: (Left) Visualization of the Husky trajectory and landmarks during the teach phase. The green path denotes the taught path. The green markers are the triangulated landmarks and the yellow markers are the refined estimates for the landmarks. Only the subset of landmarks close to the vehicle location is shown. (Right) Frame to frame feature matches used for triangulating new landmarks as well as estimating vehicle pose change

4.4 Summary

Overall, the monocular pipeline results in more uncertainty and is less robust than a stereo solution as demonstrated in [11]. This can be attributed to the larger uncertainty associated with the landmark position estimates. In terms of robustness, the monocular pipeline requires constraints on the motion of the vehicle to avoid degenerate cases such as pure rotation and even regions of high path curvature. It is also weak in situations where there is rough and uneven terrain. This causes rapid motions in the image which increases the probability of failed triangulations. Recovering from such a failure requires a manual intervention due to the initialization process.

One crucial aspect of a robust monocular solution is determining the optimal distance for a new keyframe to be dropped. Too short means landmarks are not well triangulated. Too far means fewer features are matched. This has a significant impact on the performance of the VO and hence the localization and path-tracking later on. Too little distance between keyframes results in high uncertainty in the landmark positions and too much distance results in too few triangulated landmarks. This is more pronounced when the scene is close to the camera, the frame-to-frame matches drop off quickly resulting in a smaller number of potential triangulations. This effect can be seen about 55 meters and 95 meters along the path where the vehicle was driven close to the edge of the dome.

To address the issue of failed triangulations, an automatic re-initialization procedure could be implemented. This could simply mean driving the vehicle forwards for approximately one meter after triangulation failures. A more sophisticated system which takes into account the taught path can also be explored to reduce the manual interventions.

We demonstrate a monocular navigation system working in the loop on a ground-based vehicle that is useful in indoor environments using only a single calibrated camera and wheel odometry. We make no assumptions about the structure of the scene and demonstrate the system working in a difficult real-world scenario over a distance of over 1.1 kilometers. Before deploying such a system it is important to keep in mind it is susceptible to VO failures in

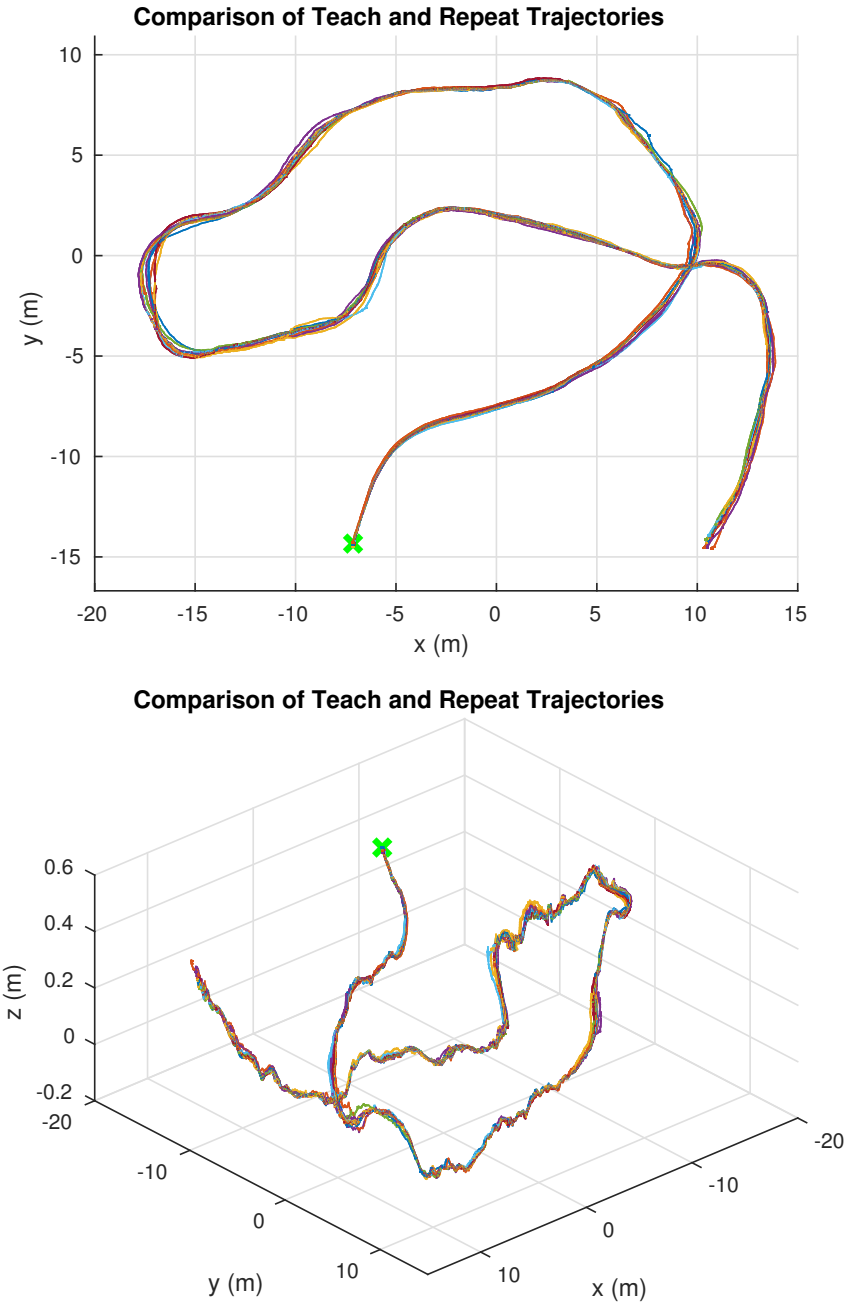

Figure 4.5: A plot of the originally taught trajectory and the 10 autonomously repeated trajectories collected using the Leica Total-Station, denoted by different colours. The left figure shows the top-down view. The right image provides a 3D view with unequal axes to highlight the z-axis. The start of the paths are denoted by the green 'X'.

Figure 4.6: Location of manual interventions at approximately 60 meters (left) and 100 meters (right) from the origin.

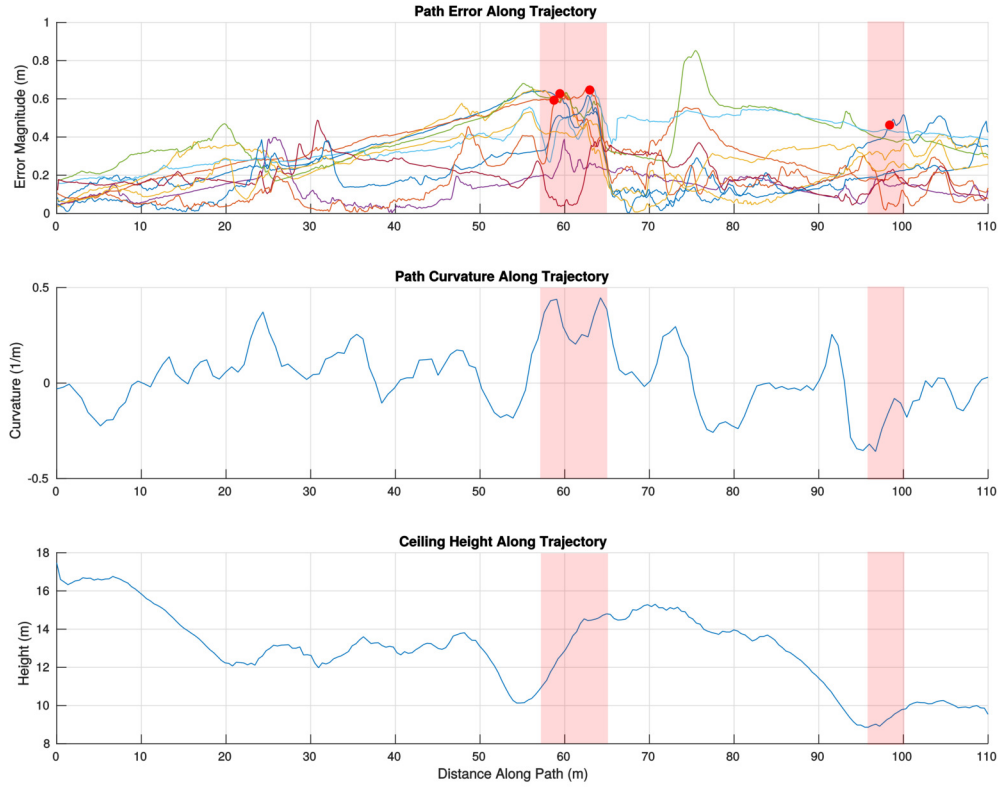

Figure 4.7: Cross track error over time for 10 repeats at a speed of 0.25 m/s (top). Four manual interventions are noted in red where the VO failed and did not recover. The vehicle had to be manually driven for approximately one meter in each case for VO re-initialization. These difficult areas are highlighted in red. They generally occur when the magnitude of path curvature is high and a rapid change in ceiling height occur in combination.

certain configurations (e.g., purely rotational motion, quick oscillation motion). Overall, with the exception of the failure cases mentioned, we are able to maintain on average 0.26 meters of tracking accuracy in a complex environment.

Chapter 5

Learning Descriptors

5.1 Overview

Descriptors are a fundamental building block used for vision-based state estimation. They must be robust to viewpoint and appearance changes while maintaining their distinctiveness so they can be re-identified. Typically, visual descriptors are developed as one-size-fits-all methods of matching, with the goal of making a descriptor as generally applicable as possible. We seek to take a tangential approach: tuning descriptors at increasing levels of specificity to a particular location and time (see Figure 5.1).

This is similar to the place-dependent features presented by McManus *et al.* [36] and Linegar *et al.* [31]. However, instead of training support vector machines (SVM) for each landmark, we use traditional binary descriptors. An evolutionary algorithm based on Generated BRIEF (GRIEF) [26] is used to learn an environment-dependent function for generating the descriptor. This allows the description function to be adapted to the appearance of the environment, tailoring it to specific scenes.

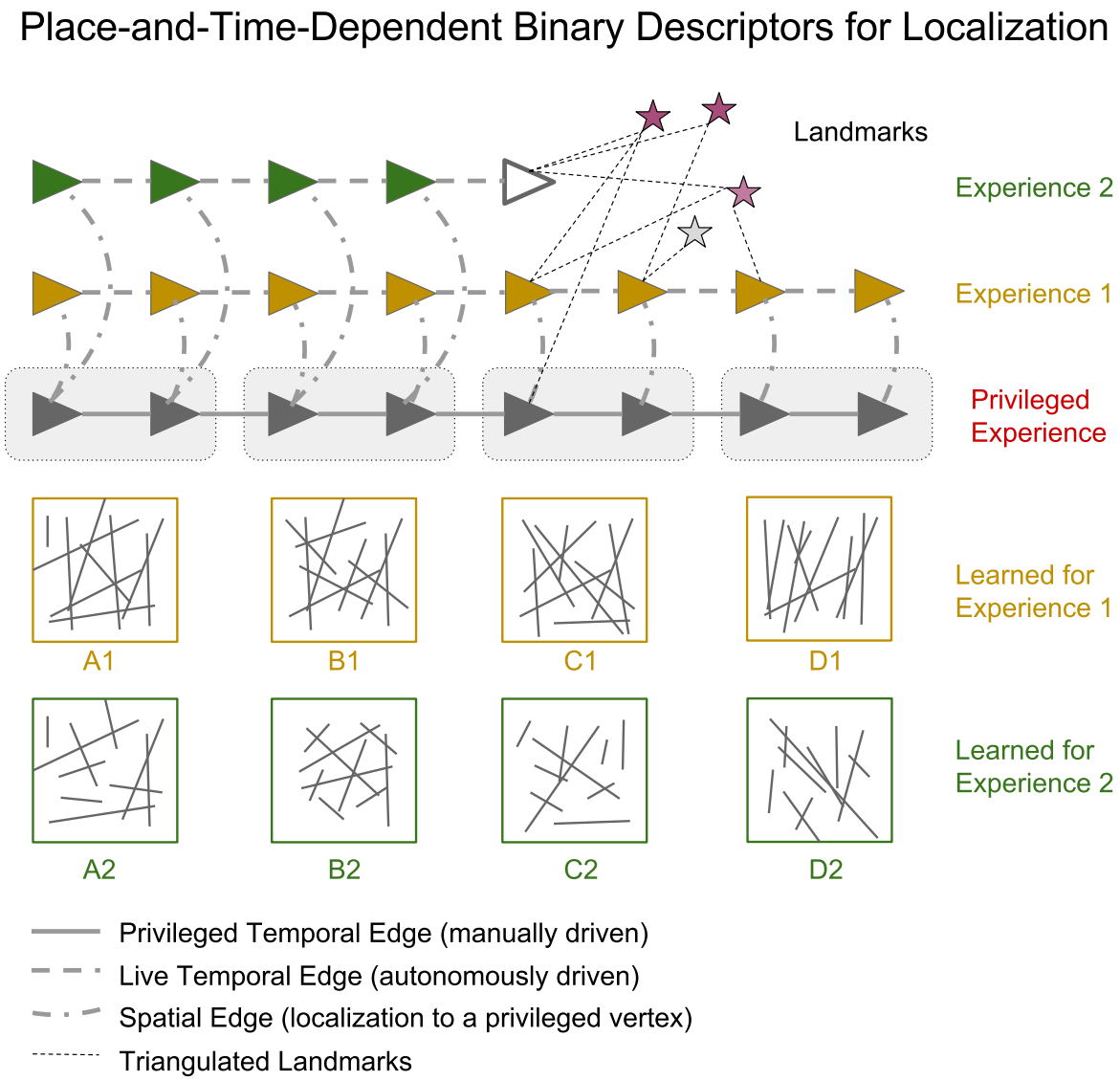

Figure 5.1: Cartoon illustration of a place-and-time-dependent feature description scheme that adapts the matching function (A, B, C, D) to a segment of the path at a certain time using binary descriptors. The vertices (triangles) represent keyframes recorded during a traversal. They are connected to each other by spatial or temporal edges containing the estimated pose. The privileged experience is the manually driven path determined to be safe by the operator. The live experience is collected during autonomous repeats. The descriptors can be trained using either only the privileged experience or multiple experiences. Correspondences generated using the adaptive descriptor results in longer and improved localization performance in the presence of scene changes.

5.2 Binary Descriptors

Given an image $\mathbf{I}$ and a keypoint of interest at $\mathbf{x}_k$, the ith bit of the descriptor can be computed from either a BRIEF comparison (5.1) or a LATCH comparison (5.2). Like Calonder *et al.* [5], we maintain a 256-bit descriptor using a fixed 48×48 pixel patch computed after applying a 9×9 box filter on the image. Each bit of the BRIEF descriptor results from an intensity comparison of two points $(\mathbf{x}_a, \mathbf{x}_b)$ with the center of the patch as the origin. Similarly, each LATCH comparison results from a comparison of the Frobenius norm between three sub-patches of size $S \times S$ pixels centered around the points $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$. For simplicity, we take the value of S to be unity as it improves the run time efficiency of the descriptor without sacrificing much performance. The comparisons are of the following form:

$$b^i_{\text{brief}}(\mathbf{I}, \mathbf{x}_k) = \mathbf{I}(\mathbf{x}_k + \mathbf{x}_a) > \mathbf{I}(\mathbf{x}_k + \mathbf{x}_b) \tag{5.1}$$

$$\begin{aligned} b^i_{\text{latch}}(\mathbf{I}, \mathbf{x}_k) &= \|\mathbf{I}(\mathbf{x}_k + \mathbf{x}_a) - \mathbf{I}(\mathbf{x}_k + \mathbf{x}_b)\| \\ &> \|\mathbf{I}(\mathbf{x}_k + \mathbf{x}_c) - \mathbf{I}(\mathbf{x}_k + \mathbf{x}_b)\| \end{aligned} \tag{5.2}$$

The intensity information varies considerably with natural scene changes. The 'gradient information' used by BRIEF and ORB is robust to some of these changes. It is reasonable to assume the 'Hessian information' used by LATCH should be more robust.

5.2.1 Data Labeling

Given only a single experience, VO matches can be used to evolve the descriptor. With multiple experiences, localization matches can also be incorporated. Both positive, S_p, and negative, S_n, correspondences are important in the evolutionary process. To obtain the set S_p, we used the estimated 6DoF pose of the vehicle, $\mathbf{T}_{ab}$, relative to an earlier vertex, and transform all the

landmarks in homogeneous coordinates, $\mathbf{p}$, back into the map frame,

$$\mathbf{p}' = \mathbf{T}_{sv}\mathbf{T}_{ab}\mathbf{T}_{sv}^{-1}\mathbf{p} \tag{5.3}$$

and then reproject them into the image plane. The transform from the vehicle frame to sensor frame is given by $\mathbf{T}_{sv}$. Any reprojected landmarks, $\mathbf{p}'$, that fall within 3 pixels of a map feature are labelled as a correspondence. These geometric correspondences, S_p, are the set of all possible matches that should have occurred given an ideal description function.

Next, we match the descriptors between the live and map images using Hamming distance. This set includes both true positive, D_{tp}, and false positive, D_{fp} matches. We can also obtain false negatives, D_{fn}, by finding elements in S_p, but not in D_{tp}. The set, S_n, is essentially equal to D_{fp}. The true negative, D_{tn}, should not matter as they do not affect the matching performance. Usually, there are far more elements in S_n compared to S_p. We find it is better to keep the two sets in roughly equal proportion, so the effect of negative correspondences does not overpower the correct correspondences.

5.3 Evolutionary Algorithm

The process of evolving the descriptor uses the genetic algorithm described in [26]. The one addition is that we filter the set, D_n, so that it is equal in size to D_p. This balances out the evolution so that it converges faster. The fitness of the ith comparison is calculated based on the sets, S_p and S_n, given in (5.4). The fitness score and inlier matches are shown in Figure 5.2 along with a visualization of the descriptor patterns during the evolution. The fitness score is important as it allows us to determine which comparisons positively contribute to the true positive matches and negatively to the false positives matches. The expectation is that as total fitness increases, the number of matches should also increase. This is true when the minimum matching threshold is set to a reasonable value. This is why we base the convergence criteria on the number of true positive matches instead of the fitness score:

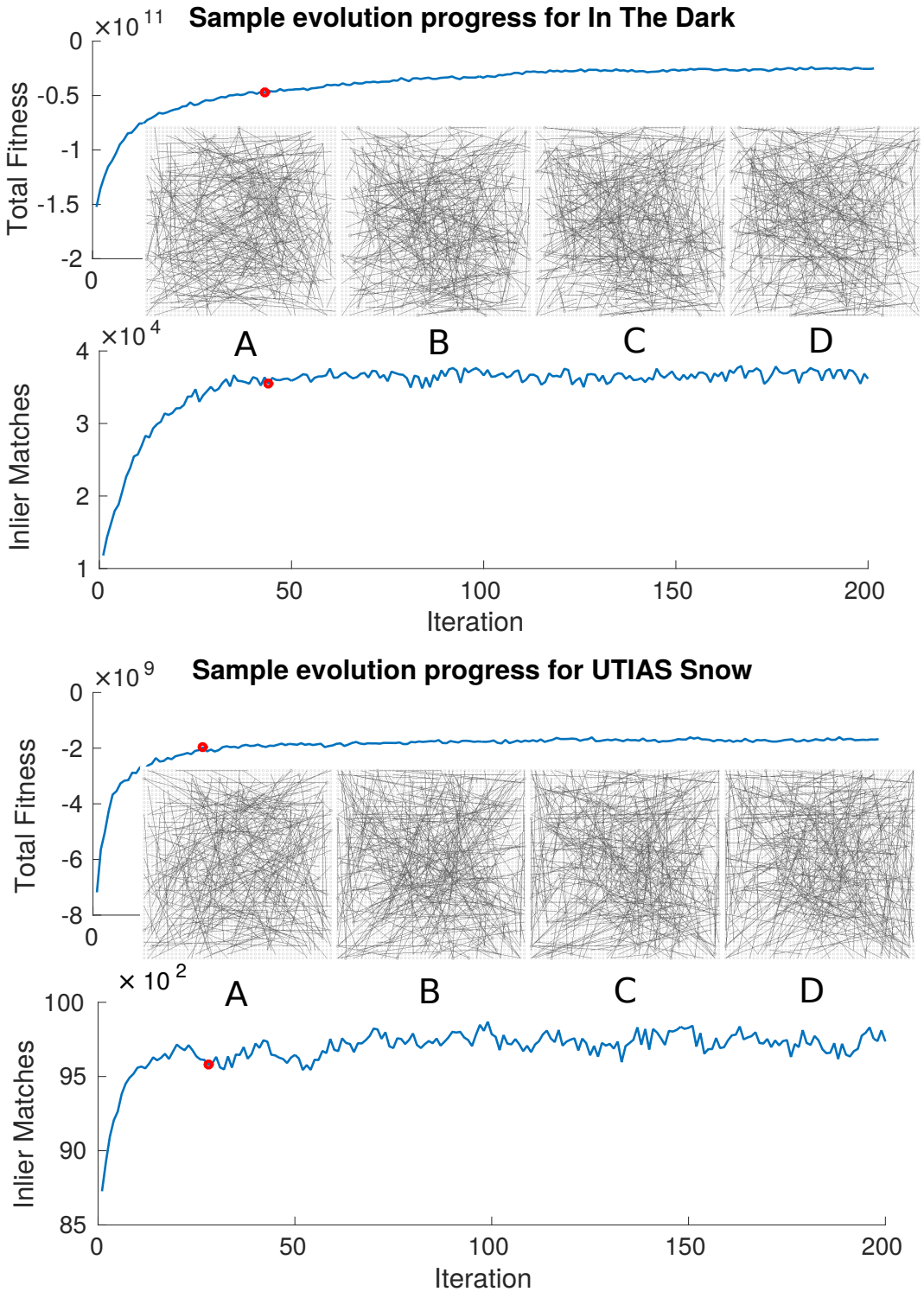

Figure 5.2: The evolution of the descriptor pattern over time for *In The Dark* and *UTIAS Snow* over 200 iterations. The total fitness asymptotically converges in both cases. The red dots denote the point at which the evolution process would be normally terminated and saved to the graph.

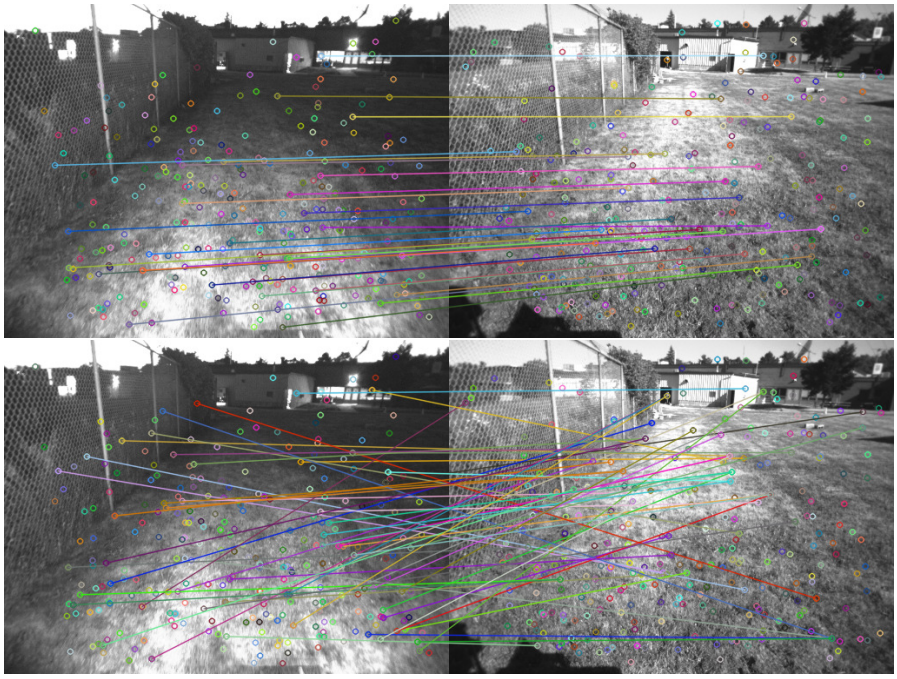

Figure 5.3: The top image shows all the possible landmark correspondences, S_p. The bottom image shows the correspondences generated using the descriptor containing the sets D_{tp} and D_{fp}. These labels can be used in the evolutionary algorithm to maximize the total fitness and therefore the number of elements in D_{tp}.

$$f_i(S_p, S_n) = \sum_{S_p}(1 - 2d_i) + \sum_{S_n}(2d_i - 1) \tag{5.4}$$

$$d_i = \begin{cases} 0, & \text{if } b^i = b'^i \\ 1, & \text{otherwise} \end{cases} \tag{5.5}$$

The evolutionary algorithm is as follows:

1. Compute all the descriptor matches from map images to live images using the current pattern

2. Re-project all live landmarks into map images using estimated transforms

3. Generate $D_{tp}, D_{tn}, D_{fp}, D_{fp}$ using geometric matches and descriptor matches

4. Add D_{tp} and D_{fn} into the set S_p, and D_{fp} into S_n
5. Filter the set S_n using a minimum matching threshold, then randomly sample it so that it is equal in size to the set S_p
6. Compute the fitness of each comparison
7. Replace the worst 20% of comparisons drawn from an uniform distribution with equal probability of either a BRIEF or LATCH comparison
8. Repeat until number of true positive matches converges or for a set number of iterations

We initialize with a random pattern drawn from a uniform distribution. Using a pre-trained pattern could lead to faster convergence. The comparison pattern for the descriptor is evolved offline using the above algorithm and written back into the corresponding vertex in the graph.

The training process takes a few minutes using an Intel i7-3720QM without any multi-threading or GPU acceleration. The maximum number of iterations is limited to 200 from experimentation, and we terminate if the number of correct matches stops increasing for 10 iterations. The authors of GRIEF trained their pattern for an hour. Presumably, we could have achieved slightly better results if we allow the algorithm to run for a longer period but this has diminishing returns. The fitness score and inlier matches are shown in Figure 5.2 along with a visualization of the descriptor pattern.

5.4 Experimental Setup

This work is presented within the VT&R system, specifically the descriptor matching portion of the localization subsystem. To isolate the performance of using different description functions along the path, we only localize back to the privileged experience. This is reflective of situations such as GPS-denied emergency return of unmanned aerial vehicles (UAV) where the scene change can be dramatic less than an hour after the original pass. It could also be beneficial in scenarios where frequent traversal of the path is difficult to achieve. The proposed

scheme should result in a more robust visual-based localization system that can wait longer periods of time before a new experience is required. Ultimately, this can be combined with multi-experience localization (MEL) [45,46] to reduce the storage and computation cost of the system.

VT&R normally uses GPU-accelerated SURF descriptors and detectors for both visual odometry (VO) as well as localization. For consistency, we maintain the use of SURF for VO, but localization is performed using the proposed environment-dependent binary descriptor. We keep the same detections from SURF for localization but re-compute the descriptors. The low computational time of binary descriptors makes it possible to achieve real-time performance.

Taking the environment-dependence idea to the extreme, a unique pattern can be used at every keyframe. We stop at the keyframe level, but one can extend this method for generating a unique pattern for parts of an image or even every landmark. This would require a change in the matching framework and could be explored in future work. Practically, learning a different descriptor for every keyframe leads to poor performance due to the small amount of training data that is available.

In The Dark deals with illumination changes and *UTIAS Snow* deals with seasonal changes. Both datasets were collected using the Clearpath Grizzly rover shown in Figure 5.6 at UTIAS. For *In The Dark*, the Grizzly was driven over the path shown in Figure 5.4 20 times over a period of 24 hours at approximately equal intervals. This totals to about 5 km of driving over both paved roads as well as grass.

For *UTIAS Snow*, the Grizzly was driven over the path shown in Figure 5.5 over 100 times from late January into early May. Only the first 50 experiences are examined as single experience localization fails past that point. Without the intermediate bridging experiences, the scene change becomes too drastic for proper landmark correspondence. This dataset is entirely over grass, but some buildings are visible. In both cases, half of the experiences are used for training E_{tr}, and the other half for testing E_{te}. Both datasets were collected autonomously using the multi-experience VT&R system as presented in [45].

Figure 5.4: (Left) Sample images for the *In The Dark* dataset. Each row shows the same location at various times during a 24-hour cycle. We see the presence of large shadows, lens flares, and poorly illuminated scenes. 20 repeats are used to validate the environment-dependent descriptor with at most half of them being used for training and the other half for testing. (Right) An aerial view of the path traversed for the *In The Dark* dataset. Each repeat totals to about 250 meters of driving around the UTIAS Dome. The first half of the path is over paved roads and the second half over grass. This path was driven approximately every hour over a span of 24 hours using multi-experience VT&R.

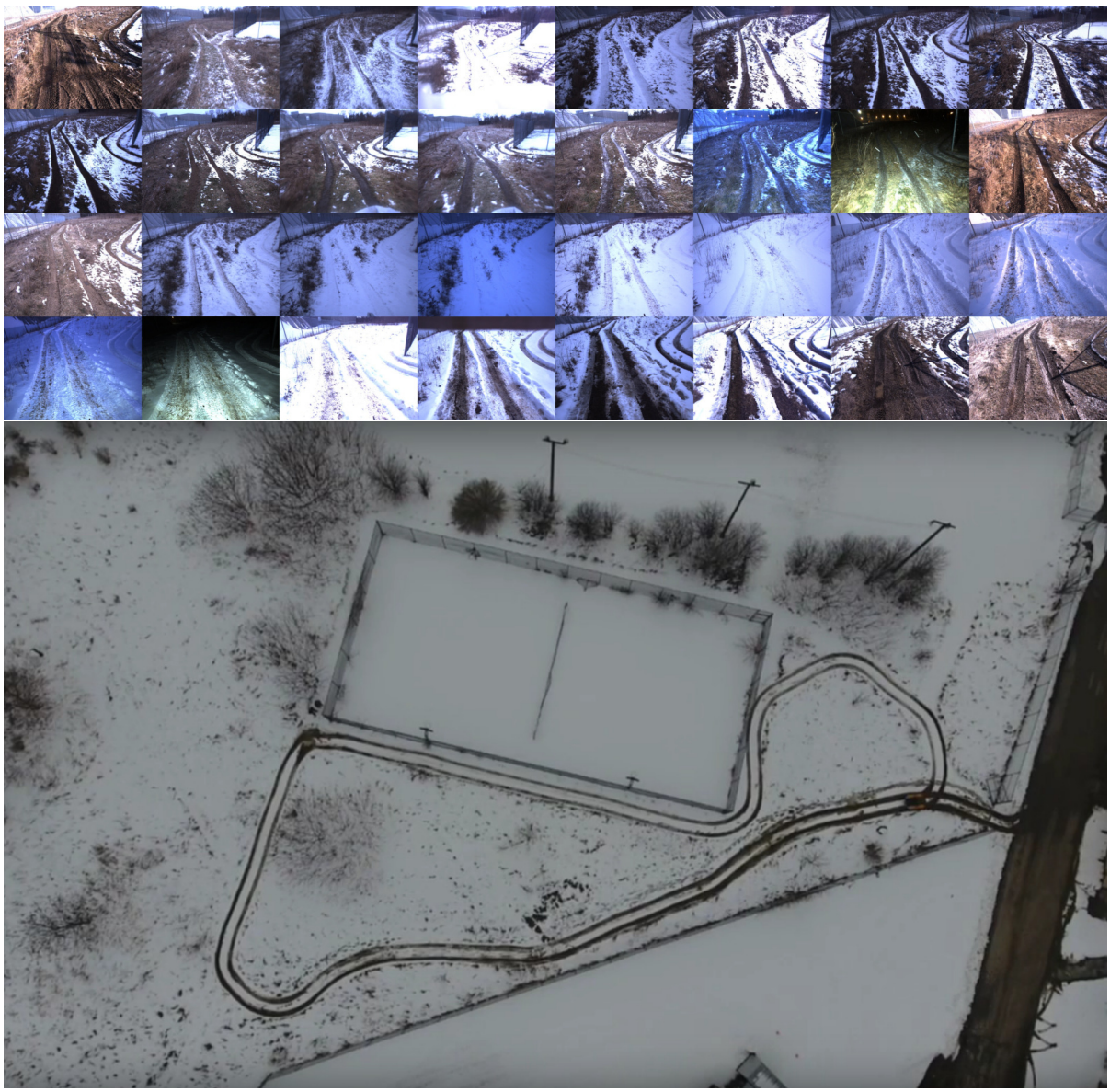

Figure 5.5: (Left) Sample images for the *UTIAS Snow* dataset. All the images show the same location at various times during the data collection process. The proposed system fails to localize when the snow completely covers the ground. About 50 repeats are used to validate the environment-dependent descriptor with at most half of them being used for training and the other half for testing. (Right) An aerial view of the path traversed for the *UTIAS Snow* dataset. Each repeat totals to about 250 meters through tall grass and rough terrain beside the tennis court at UTIAS. This path was driven at regular intervals from late January into early May using multi-experience VT&R.

Figure 5.6: The Clearpath Grizzly rover fitted with a Bumblebee XB3 stereo camera. The stereo images are logged at 10 Hz for both the *In The Dark* and *UTIAS Snow* datasets. Multi-experience localization is used to establish data correspondence.

5.5 Results

Due to differences between binary descriptors and floating-point descriptors such as SURF, we must impose different minimum matching thresholds before RANSAC. We experimentally determine the optimal thresholds that produce the most matches for both classes of descriptors. For binary descriptors, we assign a max threshold of 0.3 and SURF a value of 0.12. For binary descriptors, the value is computed as the fraction of bits that differ to the total number of bits. For SURF, it is calculated by subtracting the cosine distance from one.

We try three different schemes for training: using VO matches from the privileged experience (*pe*), using a temporally close experience from earlier in time (*se*), and using all experiences from the training set (*ae*). For each of these schemes, we also try learning a single pattern over the entire path (*s*) and learning a different one every 15 meters (*m*). This was chosen arbitrarily and splits the paths into 16 sections. Together, this creates six different scenarios: *pe-s*, *pe-m*, *se-s*, *se-m*, *ae-s*, *ae-m*. The descriptor patterns are evolved using localization results from

multi-experience VT&R in each scenario.

As an example, for *In The Dark*, the privileged (teach) experience can be considered to be *exp0*. We refer to the 20 repeats as: *exp1, exp2, . . . , exp20* in chronological order. The testing set, E_{te}, and training set, E_{tr}, correspond to the odd numbered and even numbered experiences. This means both sets contain the full 24 hours of illumination changes. We give an example of how the adaptive description pattern is generated in each case:

- pe : train on *exp0*, test on E_{te}
- se : train on *exp1*, test on *exp2*
- se : . . .
- se : train on *exp19*, test on *exp20*
- ae : train on E_{tr}, test on E_{te}

To obtain a baseline for comparison, we use the SURF descriptor and try to localize all the repeats from the test set (E_{te}) back to the privileged experience. We also do the same for other common binary descriptors such as ORB, BRIEF, and LATCH. A random pattern (*rand*) generated using a uniform distribution and the pattern that was trained in [26] (*grief*) are also tested. Finally, a hand-crafted pattern inspired by SURF is also examined (*Aster*) (see Figure 5.11). All these descriptors are compared using the six schemes noted above for both datasets.

The upper plot in Figure 5.7 shows the percentage of post-RANSAC inlier matches for each of the 10 test experiences (E_{te}) back to the map. These values are normalized based on the total number of landmarks saved during the privileged experience. The bottom plot in Figure 5.7 shows the fraction of vertices that are successfully localized. Success is defined as greater than 10 matches at a vertex.

The percentage of landmarks that can be matched drops to below 40% when repeating immediately after the teach experience. This means the majority of stored landmarks will never get matched, either because the feature detector is unable to detect them again or they

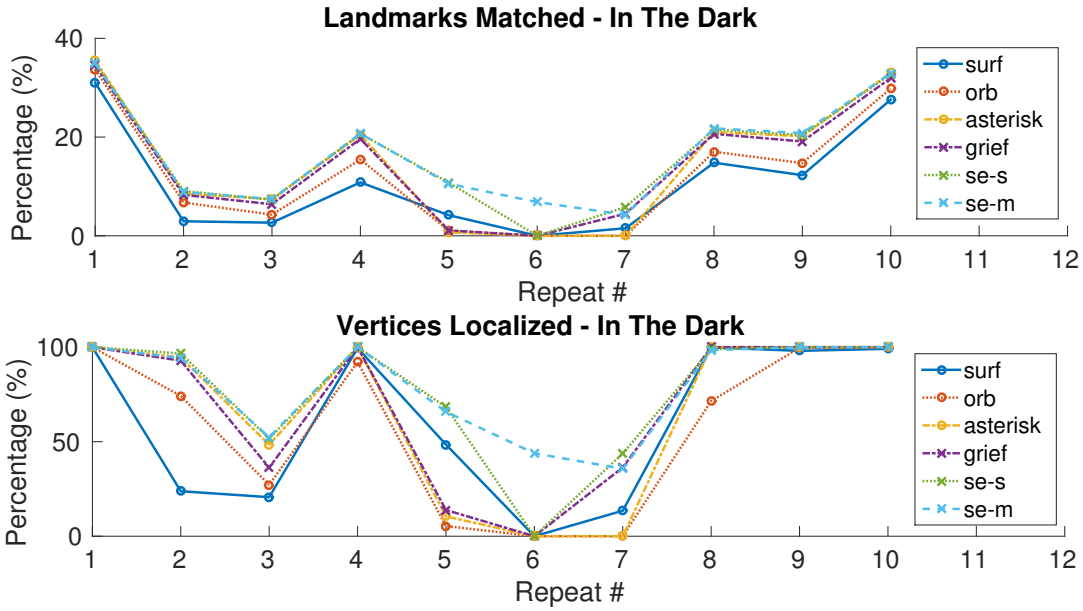

Figure 5.7: Localization results of *In The Dark* in chronological order. The top plot shows the percentage of landmarks from the privileged experience successfully matched over 10 repeats using each descriptor. The time difference between the repeats is approximately 2 hours. The bottom plot shows the percentage of vertices that were successfully localized (more than 10 matches). Only a subset of the relevant descriptors is shown.

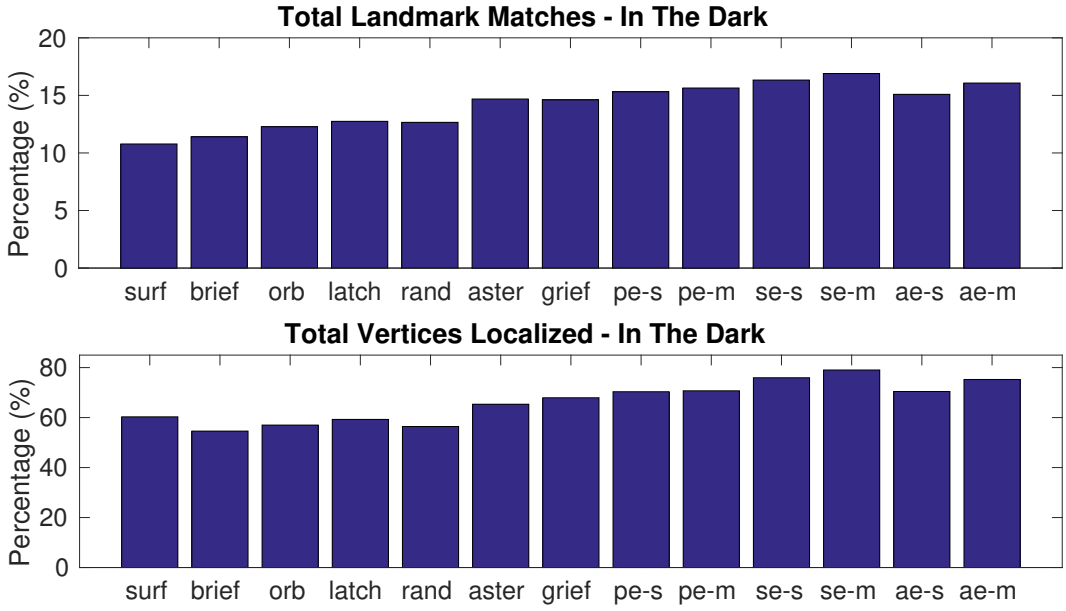

Figure 5.8: Localization results of *In The Dark* over 10 repeats. The top plot shows the total percentage of landmarks matched to the privileged experience. The bottom plot shows the total percentage of vertices localized. The evolved descriptors outperform other methods with multiple descriptors learned from similar experiences with *se-m* resulting in the best performance.

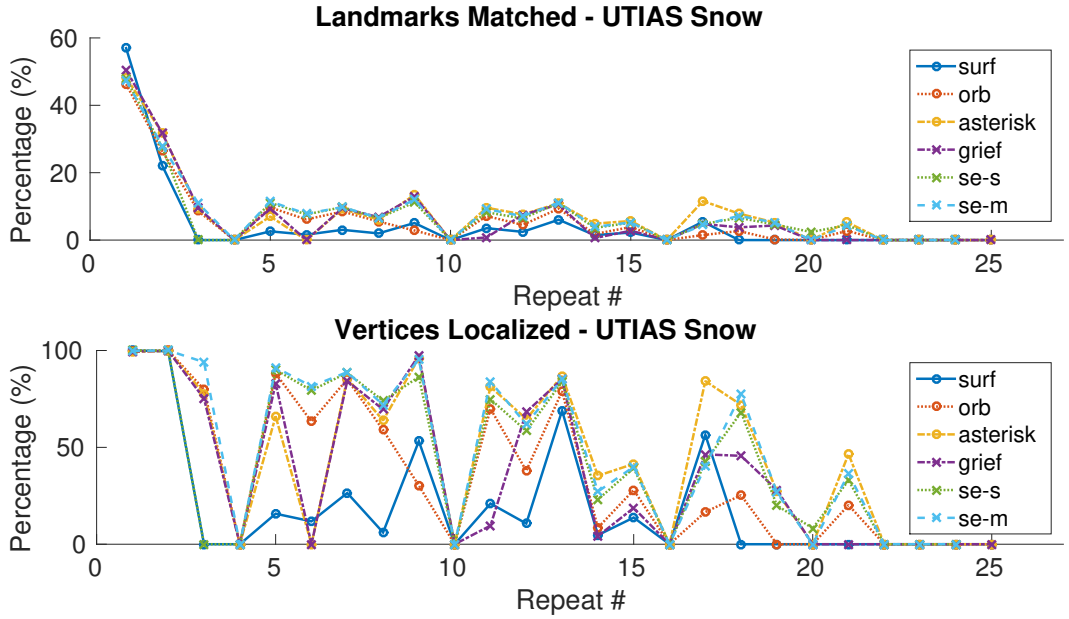

Figure 5.9: Localization results of *UTIAS Snow* in chronological order. The top plot shows the percentage of landmarks from the privileged experience successfully matched over 25 repeats using each descriptor. The time difference between the repeats is approximately every 2-3 days. The bottom plot shows the percentage of vertices that were successfully localized.

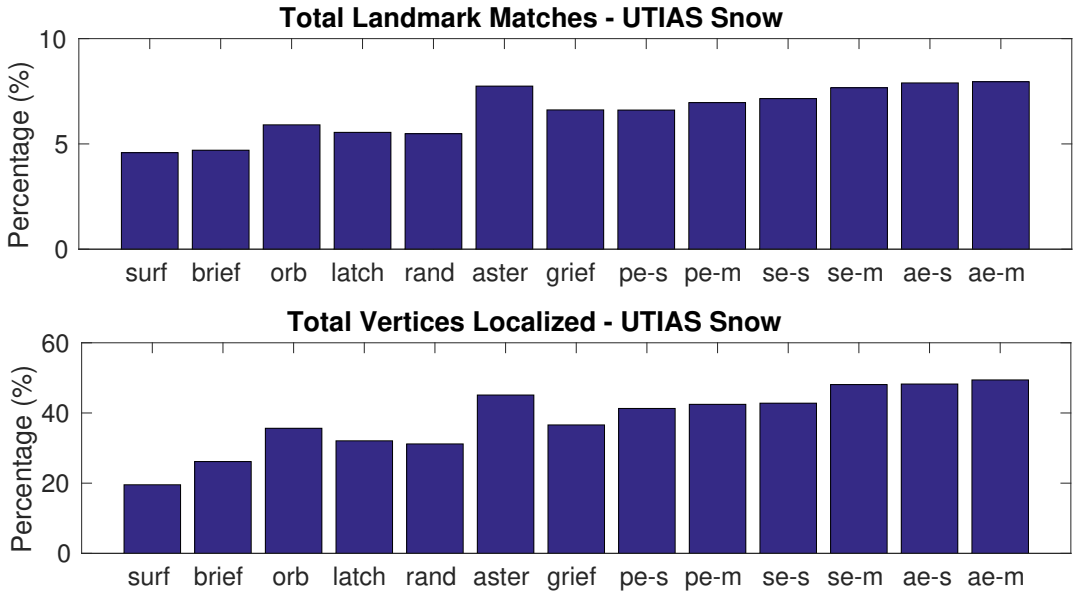

Figure 5.10: Localization results of *UTIAS Snow* over 25 repeats. The top plot shows the total percentage of landmarks matched to the privileged experience. The bottom plot shows the total percentage of vertices localized. The evolved descriptors outperform other methods with multiple descriptors learned from all experiences *ae-m* resulting in the best performance.

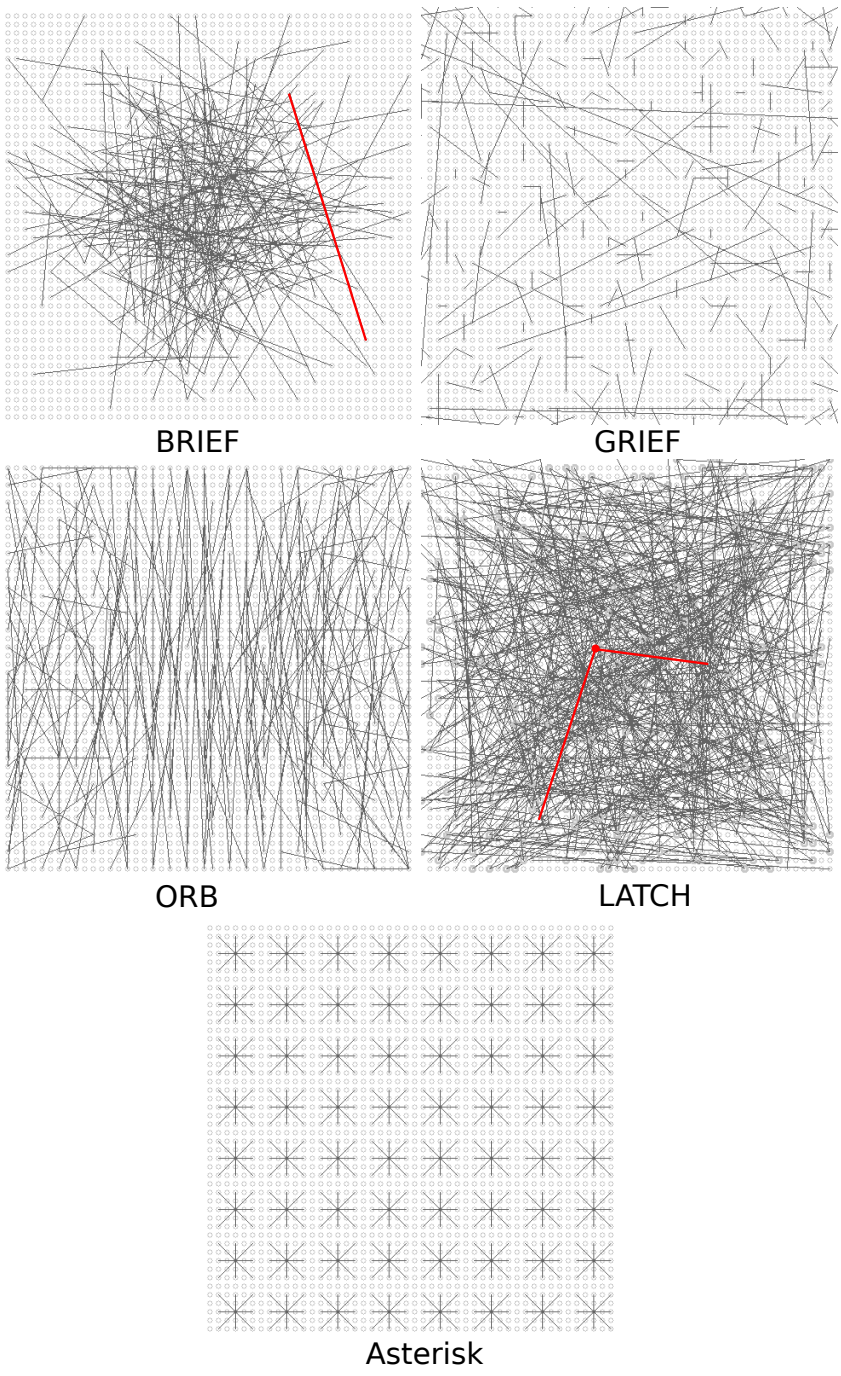

Figure 5.11: The patterns used for BRIEF, ORB, GRIEF, and LATCH. For the BRIEF inspired descriptors, the pixel comparisons are displayed as a line connecting the pixel being compared. For LATCH, the positions of the three sub-patches are connected using two lines with the center position denoted using a bold circle. An example of one comparison in each case is highlighted in red. A hand-crafted descriptor (Aster) inspired by SURF and the results from GRIEF is also shown.

are not distinctive enough. A feature detector that can consistently produce the same detections is crucial to localization performance.

During repeats 2 and 3, the number of matches drops due to the presence of long shadows and lens flares. Repeats 5, 6, and 7 correspond to nighttime repeats. Coming back to the same time the next day, the number of matches increases back to around 40%. It is important to note that the only scheme that produces matches during repeat 6 is *se-m*, corresponding to using different description functions along the path trained using visually similar experiences.

Examining the results at a higher level in Figure 5.8, we see the learned descriptors outperform the traditional descriptors, increasing the percentage of localizable vertices from around 60% to 75%. Using multiple descriptors along the path results in slightly improved localization results across the board, *pe-m, se-m, ae-m*. By changing the comparison patterns along the path, it restricts the range of the description function, making it more discriminative to the visual information at specific locations. As expected, training the descriptor using visually similar experiences results in the best performance (*se-m*).

Notably, training using only the privileged experience produces a similar matching performance to training using all experiences. In this case, it means the evolution is mainly increasing the robustness of the descriptor to viewpoint changes. Binary descriptors effectively handle large illumination change by design.

Similar improvements are seen for the *UTIAS Snow* dataset, shown in Figure 5.9 and 5.10. The effect of using similar experiences for training is less effective than the other schemes compared to the results obtained with *In The Dark*. This is likely due to the substantial physical changes in the environment during successive experiences. By training on specific experiences, the evolutionary algorithm allows the description function to over-fit to the location. This is not a problem for illumination changes due to the robustness of binary descriptors in that particular case. However as the scene physically changes, this over-fitting becomes problematic.

Compared to SURF, the percentage of localizable path increases from 20% to close to 50%. The number of matches increases by more than 70%. The performance fluctuates as snow falls

and melts. In repeats 4, 10, and 16, localization fails over the entire path. A significant amount of snowfall was accumulated after repeat 22 and the system was no longer able to localize.

It is interesting to note the GRIEF pattern shown in Figure 5.11 does extremely well in our dataset. This may be attributed to the shorter comparison patterns that were observed by the authors of GRIEF. Motivated by this and taking inspiration from the sub-regions used in SURF we create a hand-crafted binary pattern called Aster (see Figure 5.11). It performs exceptionally well on the snow dataset coming very close to the performance of the evolved descriptors. It can be used as an initial pattern for evolution or simply as is. This demonstrates that certain patterns are better than others for localization and a single pattern does not necessarily generalize to all environments, hence the proposed system. It would be interesting to see if the matching performance of Aster holds up in other types of environments.

5.6 Summary

We presented an unsupervised method of feature matching using learned place-and-time-dependent descriptors. It is demonstrated that this increases the localization ability of single-experience VT&R while maintaining similar computational complexity. We demonstrate day-to-night localization without the use of expensive low-light cameras and pre-processing of the images, which will further improve localization performance. In the case of extreme environmental changes, the representational power given by binary descriptors is insufficient for long-term operation. However, we do see improved matching and localization performance compared to other descriptors.

The performance of the proposed method is affected by the training data used. For testing, we set a fixed interval for switching to a new descriptor and tried a variety of training strategies. It is best to use all the training data that is available, but an intelligent method of determining how often to learn a new descriptor along a path is crucial for optimal matching performance. It is a trade-off between the generality of the descriptor across scene changes and its specificity

to a particular location and time.

A major issue with the current implementation of the place-and-time-dependent descriptor is that we still rely on the SURF detector. As the scene change become more drastic, the ground truth correspondences also decrease due to the detector not firing at the same locations. This becomes a bottleneck for the evolutionary algorithm as without the properly labeled examples of feature correspondences the evolution is unable to make progress.

An interesting extension might be to replace the description function with a neural network. This could offer much more representational power and could be trained using a Siamese network. The inference time for a shallow multilayer perceptron is computationally cheap especially with GPUs and allow it to handle not only illumination but seasonal changes as well. With the addition of convolutional layers, it could start to learn the appearance of dominant landmarks as more experience is gathered.

Chapter 6

In the Loop Demonstration

6.1 Overview

This chapter combines the previous work from chapter 4 and chapter 5 to demonstrate the monocular system working online using the time-and-place-dependent descriptors (PDD). The same hardware setup was used as in chapter 4 with the exception of the camera orientation (Figure 6.1). To analyze the performance of the system in the presence of lighting changes, an outdoor setting was chosen. Due to this change in setting, the camera is pointed downwards towards the ground as opposed to upwards toward the sky.

A short path of around 20 meters was chosen as a demonstration of the localization performance over a half day period from 10 am to 8 pm. Each repeat is performed approximately an hour apart, totaling 10 repeats and about 200 meters of driving. An example of the terrain and lighting can be seen in Figure 6.2. It is important to note the algorithm does not rely upon the cones placed along the path as demonstrated in chapter 4, they were simply added as a visual confirmation for proper path following.

In order to compare all descriptors on even footing we used the hand-crafted Aster descriptor within the place-and-time-dependent localization module for all the live repeats to obtain the labeled data. Then we run the other descriptors offline on the same data to compare per-

Figure 6.1: The Clearpath husky rover with a Stereo Lab Zed camera and Lenovo P50 laptop.

Figure 6.2: Example of the lighting change over the testing period. The orange cones are used as an visual feedback to ensure the vehicle is not deviating off the taught path. The VT&R algorithm is not dependent on them for proper path following.

formance. As demonstrated previously the best results come from training on temporally close experiences with similar conditions as the current conditions. This configuration is chosen for the adaptive place-dependent descriptor scheme and is compared to both Aster and SURF.

6.2 Results

We examine the localization performance of the system over the 10 repeats. The key metric is the post-RANSAC inlier matches of the different descriptor schemes over all the repeats. The results are very similar to chapter 5 as expected. We see an 50% increase in inlier matches going from SURF to ASTER and 54% to the learned descriptor (PDD) (Figure 6.3 and Table 6.1).

Repeat	SURF	ASTER	PDD
1	7893	9538	10226
2	6791	8847	9493
3	4621	6857	7134
4	4241	6557	6757
5	3795	6169	6091
6	4081	6600	6656
7	3555	5807	5763
8	2729	5031	5073
9	3464	6037	5871
10	3647	5899	6024
Total	44817	67342	69088

Table 6.1: Total number of landmarks correspondences using different descriptor schemes.

6.3 Summary

The results presented in the section confirms the results presented in chapter 5. Using the adaptive descriptor scheme shows a significant increase in performance over the traditional descriptors. The handcrafted descriptor ASTER also performs on par with the learned descriptor. It should be noted this is only true for this specific dataset. As shown previously it does not

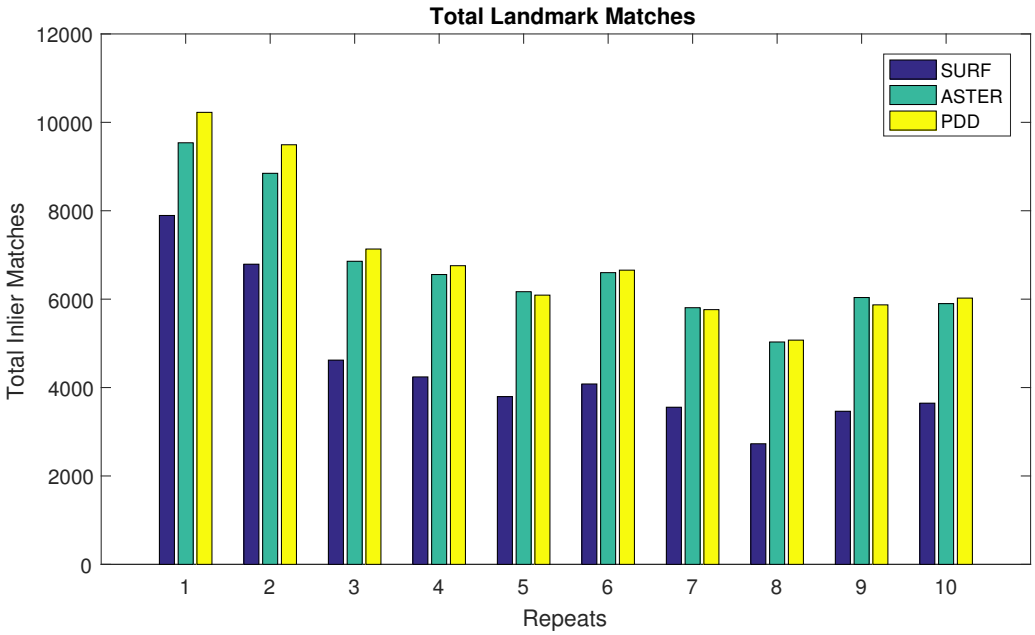

Figure 6.3: Bar graph of total post-RANSAC inlier correspondences resulting from SURF, ASTER, and PDD over the 10 repeats. Similar increase in performance is demonstrated compared to the offline data from the In The Dark and UTIAS Snow datasets. ASTER proves to be extremely well performing on this dataset, likely due to the mostly grass environment.

necessarily generalize to all environments, hence the need for the learning process.

Changing the camera orientation and environment did present some problems for localization. As soon as localization is lost it is unlikely to recover as VO is not as reliable as in the monocular case. Relaxing the matching requirements for localization helps, but ultimately we are limited by the uncertainty in the VO and wheel odometry.

The wheel odometry measurements are not enough to constrain the scale drift inherent to a monocular solution. This was a not a huge issue in the case of the dome because the scene was sufficiently far and even large changes in scale still result in good localization. This is not the case when the scene is close as in the case with the outdoor tests. Over a longer distance the scale drift becomes problematic. A setup with a better secondary sensor is necessary to correct this issue.

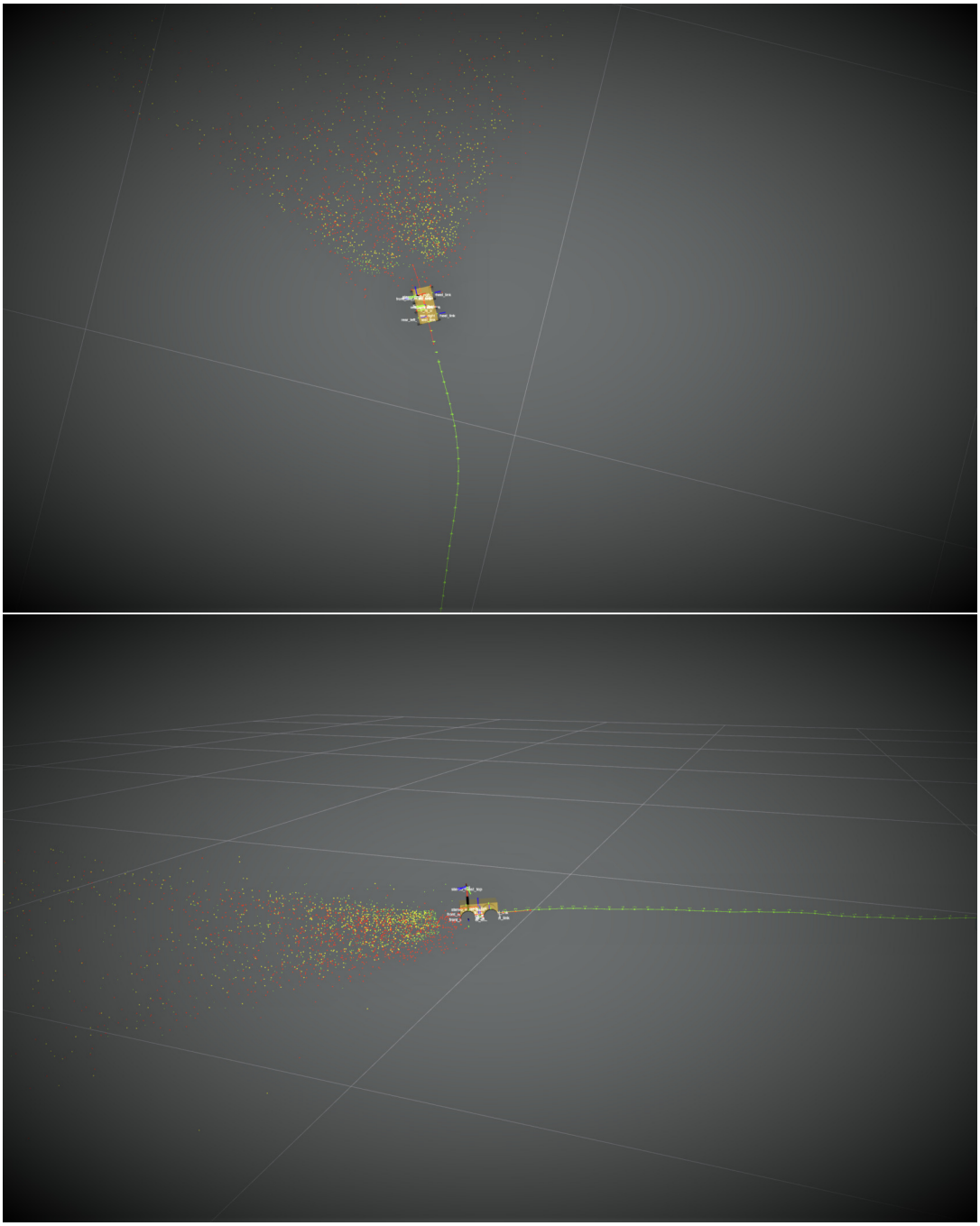

Figure 6.4: Visualization of landmarks being tracked, majority of landmarks are from the ground.

Chapter 7

Conclusion

7.1 Summary

In this thesis, we presented a monocular system that is able to learn from previous experiences in order to generate better feature correspondences. The monocular VT&R solution provides a simple, cost-effective solution for robot navigation especially in indoor environments. No assumptions about the structure of the scene are required. We present experimental validation of the system successfully repeating approximately 1.1 km of driving with an autonomy rate of 99.6%. In certain camera configurations, the scale drift poses more of a problem than others, especially when the scene is close to the camera as is the case in Chapter 6. Significant manually tunning of parameters is required to achieve proper path following.

Under the motivation of extending the duration of time between map creation and vision-based localization, we explore a learning-based method using an evolutionary algorithm. This is well suited for repeating the same path over and over again in the presence of scene changes. We also presented a handcrafted descriptor ASTER which exhibits similar performance to the result of the learned descriptors in specific scenarios. This method drastically increases the number of feature correspondences by at least 40% comparing to classical descriptors such as SURF, BRIEF, and ORB. It also increases the number of successful localizations by at least

Figure 7.1: Using sematic information to aid long term localization. (Top) Example of objector detector using YOLO [50], (Bot) Example of pixel-wise segmentation using SegNet [2].

25% in the datasets presented.

The monocular VT&R implementation with real-world results as presented in Chapter 4 has been published and peer reviewed in [39]. The novel learning-based descriptor scheme presented in Chapter 5 has been published and peer reviewed in [40].

7.2 Future Work

As discussed in Chapter 5, a major issue is the ability of the detector to repeated fire on the same location in the presence of scene changes. There are two approaches that could potentially resolve this issue: learning a detector or do not use the detector at all. In the first case, an example would be a method that combines both the detector and descriptor scheme into a single learning-based system. It would simultaneously learn both the optimal detection function as

well as the description function. An example of this is LIFT, using a deep neural network to handle the entire feature detector and description pipeline. Other image transforms such as seasonal image transforms are also another interesting avenue of research.

Alternatively, it is possible to continue using the binary descriptor due to its computational efficiency and simply compute the descriptor at every pixel in the images [33]. This brute force approach will remove the need to use a detector and provide even more labeled data for training. This can eventually lead to more sophisticated dense methods for metric localization.

An interesting avenue of research is integrating semantic information into the localization process. This is made possible by recent advancements in the area of object detection and classification. Some examples of popular networks include You Only Look Once (YOLO), SSD and Fast Region-based Convolutional Neural Network (R-CNN) [32, 50, 51]. Good performance is also being observed from per-pixel classification networks [2, 8]. With these machine learning approaches, repeated and consistent detection of stable large-scale feature in the environment (roads, building, signs, trees) is becoming a possibility. Together with the learning-based feature approach presented here, it should be possible to create a much more robust localization and mapping system in a cost-effective and scalable manner.

Bibliography

[1] Sean Anderson and Timothy D Barfoot. Full STEAM ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on SE(3). In *Intelligent Robots and Systems (IROS)*, pages 157–164, sep 2015.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[3] Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2017.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision*, 2006.

[5] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[6] Nicholas Carlevaris-Bianco and Ryan M Eustice. Learning visual feature descriptors for dynamic lighting conditions. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2769–2776. IEEE, 2014.

[7] Sarah Huiyi Cen and Paul Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[9] Winston Churchill and Paul Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4525–4532. IEEE, 2012.

[10] Lee E. Clement and Jonathan Kelly. How to train a CAT: learning canonical appearance transformations for robust direct localization under illumination change. *CoRR*, abs/1709.03009, 2017.

[11] Lee E Clement, Jonathan Kelly, and Timothy D Barfoot. Monocular Visual Teach and Repeat Aided by Local Ground Planarity. In David Wettergreen and Timothy Barfoot, editors, *Field and Service Robotics*, chapter VI, pages 547–561. Springer International Publishing, Toronto, 2015.

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

[13] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[14] Feras Dayoub and Tom Duckett. An adaptive appearance-based map for long-term topological localization of mobile robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3364–3369. IEEE, 2008.

[15] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[16] Jakob Engel, Thomas Sch, and Daniel Cremers. Direct Monocular SLAM. pages 834–849, 2014.

[17] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *Robotics: Science and Systems*, 2015.

[18] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Transactions on Robotics*, 33(2):249–265, 2017.

[19] Paul Furgale and Timothy D Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

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

[21] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOSA A*, 5(7):1127–1135, 1988.

[22] Seo-Yeon Hwang and Jae-Bok Song. Monocular vision-based slam in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 58(10):4804–4812, 2011.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016.

[24] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10. Ieee, nov 2007.

[25] Kurt Konolige and Motilal Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[26] Tomáš Krajník, Pablo Cristóforis, Keerthy Kusumam, Peer Neubert, and Tom Duckett. Image features for visual teach-and-repeat navigation in changing environments. *Robotics and Autonomous Systems*, 88:127–141, 2017.

[27] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

[28] Vincent Leptit, Fransec Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155, 2009.

[29] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[30] Chris Linegar, Winston Churchill, and Paul Newman. Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 90–97. IEEE, 2015.

[31] Chris Linegar, Winston Churchill, and Paul Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 787–794. IEEE, 2016.

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

[33] Miguel Bordallo López, Alejandro Nieto, Jani Boutellier, Jari Hannuksela, and Olli Silvén. Evaluation of real-time lbp computing in multiple architectures. *Journal of Real-Time Image Processing*, 13(2):375–396, 2017.

[34] Kirk MacTavish, Michael Paton, and Timothy D Barfoot. Beyond a shadow of a doubt: Place recognition with colour-constant images. In *Field and Service Robotics*, pages 187–199. Springer, 2016.

[35] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 901–906. IEEE, 2014.

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

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

[38] Raul Mur-Artal, J. M.M. Montiel, and Juan D. Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[39] Zhang N, Warren M, , and Barfoot T D. Eye on the sky: An upward-looking monocular teach-and-repeat system for indoor environments. In *Computer and Robot Vision (CRV), 15th Conference on*. CIPPRS, 2018.

[40] Zhang N, Warren M, , and Barfoot T D. Learning place-and-time-dependent binary descriptors for long-term visual localization. In *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018.

[41] Peer Neubert, Niko Sunderhauf, and Peter Protzel. Appearance change prediction for long-term navigation across seasons. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 198–203. IEEE, 2013.

[42] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[43] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20, jan 2006.

[44] Chris J Ostafew, Angela P Schoellig, and Timothy D Barfoot. Robust constrained learning-based nmpc enabling reliable mobile robot path tracking. *The International Journal of Robotics Research*, 35(13):1547–1563, 2016.

[45] Michael Paton, Kirk MacTavish, Michael Warren, and Timothy D Barfoot. Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1918–1925. IEEE, 2016.

[46] Michael Paton, François Pomerleau, and Timothy D. Barfoot. *In the Dead of Winter: Challenging Vision-Based Path Following in Extreme Conditions*, pages 563–576. Springer International Publishing, Cham, 2016.

[47] Eduardo Perdices, Luis M López, and José M Canas. Lineslam: Visual real time localization using lines and ukf. In *ROBOT2013: First Iberian Robotics Conference*, pages 663–678. Springer, 2014.

[48] Andreas Pfrunder, Angela P. Schoellig, and Timothy D. Barfoot. A Proof-of-Concept Demonstration of Visual Teach and Repeat on a Quadrocopter Using an Altitude Sensor and a Monocular Camera. In *Conference on Computer and Robot Vision (CRV)*, pages 238–245, 2014.

[49] Horia Porav, Will Maddern, and Paul Newman. Adversarial training for adverse conditions: Robust metric localisation using appearance transfer. *CoRR*, abs/1803.03341, 2018.

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

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

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

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

[54] Jianbo Shi et al. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[55] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.

[56] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014.

[57] Hauke Strasdat, Andrew J Davison, JM Martìnez Montiel, and Kurt Konolige. Double window optimisation for constant time visual slam. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2352–2359. IEEE, 2011.

[58] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2.

[59] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction.

[60] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B Cremers, Frank Dellaert, Dieter Fox, Dirk Hahnel, Charles Rosenberg, Nicholas Roy, Jamieson Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*, volume 3. IEEE, 1999.

[61] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics (intelligent robotics and autonomous agents series).

[62] P Torr. MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*, 78(1):138–156, apr 2000.

[63] Phil Torr. An Assessment of Information Criteria for Motion Model Selection. In *Computer Vision and Pattern Recognition*, pages 47–52, San Juan, 1997. IEEE.

[64] Tommi Tykkala, Andrew I. Comport, and Joni Kristian Kamarainen. Photorealistic 3D mapping of indoors by RGB-D scanning process. *IEEE International Conference on Intelligent Robots and Systems*, (November):1050–1055, 2013.

[65] Christoffer Valgren and Achim J Lilienthal. Sift, surf and seasons: Long-term outdoor localization using local features. In *EMCR*, 2007.

[66] Michael Warren, Michael Paton, Kirk MacTavish, Angela P. Schoellig, and Timothy D. Barfoot. Towards Visual Teach & Repeat for GPS-Denied Flight of a Fixed-Wing UAV. *Field and Service Robotics*, pages 1–14, 2017.

[67] Stephen Williams and Ayanna M. Howard. Developing monocular visual pose estimation for arctic environments. *Journal of Field Robotics*, 27(2):145–157, 2010.

[68] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[69] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.