Expanding the Limits of Vision-Based Autonomous Path Following

by

Michael Paton

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
University of Toronto Institute for Aerospace Studies

Abstract

Expanding the Limits of Vision-Based Autonomous Path Following

Michael Paton
Doctor of Philosophy
University of Toronto Institute for Aerospace Studies
2018

Autonomous path-following systems allow robots to traverse large-scale networks of paths using on-board sensors. These methods are well suited for applications that involve repeated traversals of constrained paths such as factory floors, orchards, and mines. Through the use of inexpensive, commercial, vision sensors, these algorithms have the potential to enable robotic applications across multiple industries. However, these applications will demand algorithms capable of long-term autonomy. This poses a difficult challenge for vision-based systems in unstructured and outdoor environments, whose appearances are highly variable. While techniques have been developed to perform localization across extreme appearance change, most are not suitable or untested for vision-in-the-loop systems such as autonomous path following, which requires continuous metric localization to keep the robot driving. This thesis extends the performance of vision-based autonomous path following through the development of novel localization and mapping techniques. First, we present the following generic localization frameworks: i) a many-to-one localization framework that combines data associations from independent sources of information into single state-estimation problems, and ii) a multi-experience localization and mapping system that provides metric localization to the manually taught path across extreme appearance change using bridging experiences gathered during autonomous operation. We use these frameworks to develop three novel autonomous path-following systems: i) a lighting-resistant system capable of autonomous operation across daily lighting change through the fusion of data from traditional-grayscale and color-constant images, ii) a multi-stereo system that extends the field-of-view of the algorithm by fusing data from multiple stereo cameras, and iii) a multi-experience system that uses both localization frameworks to achieve reliable localization across appearance change as extreme as night vs. day and winter vs. summer. These systems are validated through a collection of extensive field tests covering over 213 km of vision-in-the-loop autonomous driving across a wide variety of environments and appearance change with an autonomy rate of 99.7% of distance traveled.

Dedication

To my lovely and brilliant wife, Holly.
Thank you for all of the support over the years.
This thesis would not have been possible without you.

Acknowledgements

This thesis would not have been possible without the support of my friends, family and colleagues. First and foremost, I would like to thank my parents and sister for their unconditional support over the years for which I am forever grateful.

I would also like to extend my deepest gratitude to my graduate supervisors, Professor Jana Kosecka and Professor Timothy D. Barfoot. Jana, thank you for your encouragement to pursue a master's degree, your excellent guidance over the years, and your advice to publish to the Computer and Robot Vision (CRV) conference in Toronto, where I had a chance meeting with Professor Barfoot. Tim, thank you for the once in a lifetime opportunity to join your lab and advance my career in robotics. Furthermore, thank you for sharing your knowledge of state estimation and providing me with fascinating and fun challenges to tackle during our time together. Lastly, thank you for introducing me to the joys of field robotics. The field deployments were without a doubt the highlight of my time in the lab, and I intend to continue testing robots in their natural habitat as much as possible.

My interests in robotics and computer vision began during my undergraduate studies at George Mason University. Thank you Professors Sean Luke, Jana Kosecka, and Zoran Duric for your outstanding classes on the subjects, they were what motivated me to return to graduate school and pursue a career in robotics.

I would like to acknowledge all of my colleagues that I had the pleasure of working with both at the Autonomous Space Robotics Laboratory (ASRL) and elsewhere. Many thanks to the members of the Visual Teach & Repeat (VT&R) 2.0 development team, Kirk MacTavish, Kai Van Es, Michael Warren, Katarina Kujic, and Peter Berczi, it was a great pleasure to work with you all and see our integrated code running in the field. I would like to thank Chris Ostafew, Jonathan Gammel, and Sean Anderson for bringing me up to speed on VT&R. To Patrick McGarey, thanks for all of the shared adventures in Toronto and I look forward to working with you at the Jet Propulsion Laboratory (JPL). I would like to also acknowledge Horia Porav from Oxford University for his assistance in performing a comparison of algorithms, I owe you a debt of gratitude.

Finally I would like to acknowledge the NSERC Canadian Field Robotics Network (NCFRN) and Clearpath robotics for providing the funding that made this research possible.

Contents

List of Tables

List of Figures

List of Acronyms

6DOF six-degree-of-freedom

ASRL Autonomous Space Robotics Laboratory

BoW Bag of Words

CC Color-Constant

CDF Cumulative Distribution Function

CF Collaborative Filtering

CLAHE Contrast Limited Adaptive Histogram Equalization

CRV Computer and Robot Vision

CSA Canadian Space Agency

DCNN Deep Convolutional Neural Network

DCS Dynamic Covariance Scaling

DRDC Defence Research and Development Canada

EBN Experience-Based Navigation

EKF Extended Kalman Filter

FSR Field and Service Robotics

GP Gaussian Process

GPS Global Positioning System

ICRA International Conference on Robotics and Automation

IROS Intelligent Robots and Systems

IMU Inertial Measurement Unit

JFR Journal of Field Robotics

JPL Jet Propulsion Laboratory

LiDAR Light Detection and Ranging

LSD Large Scale Direct

MAP Maximum A Posteriori

MATS Multi-Agent Tactical Sentry

MAV Micro Aerial Vehicle

MCL Multi-Channel Localization

MEL Multi-Experience Localization

MET Mars Emulation Terrain

MER Mars Exploration Rover

MCPTAM Multi-Camera Parallel Tracking and Mapping

MSL Mars Science Laboratory

NDT Normal Distribution Transform

NEES Normalized Estimation Error Squared

NID Normalized Information Distance

NIS Normalized Innovation Squared

NRP Network of Reusable Paths

NCFRN NSERC Canadian Field Robotics Network

ORI Oxford Robotics Institute

PoaP Photocopy of a Photocopy

PTAM Parallel Tracking and Mapping

PGR Point Grey Research

RANSAC RANdom SAmple Consensus

RMS Root Mean Squared

RMSE Root Mean Squared Error

RRT Rapidly-exploring Random Tree

RUV Robotic Utility Vehicle

SLAM Simultaneous Localization and Mapping

STEAM Simultaneous Trajectory Estimation and Mapping

STPG Spatio-Temporal Pose Graph

SURF Speeded Up Robust Features

SVM Support Vector Machine

ToD Time of Day

USAC Universal Framework for RANSAC

UTIAS University of Toronto Institute for Aerospace Studies

VT&R Visual Teach & Repeat

VO Visual Odometry

Notation

– General Notation –

a	This font is used for quantities that are real scalars
$\mathbf{a}$	This font is used for quantities that are real column vectors
$\mathbf{A}$	This font is used for quantities that are real matrices
$\boldsymbol{A}$	This font is used for time-invariant system quantities
$p(\mathbf{a})$	The probability density of $\mathbf{a}$
$p(\mathbf{a}\|\mathbf{b})$	The probability density of $\mathbf{a}$ given $\mathbf{b}$
$\mathcal{N}(\mathbf{a}, \mathbf{B})$	Gaussian probability density with mean $\mathbf{a}$ and covariance $\mathbf{B}$
$\mathcal{GP}(\boldsymbol{\mu}(t), \boldsymbol{\mathcal{K}}(t, t'))$	Gaussian process with mean function, $\boldsymbol{\mu}(t)$, and covariance function, $\boldsymbol{\mathcal{K}}(t, t')$
$(\cdot)_k$	The value of a quantity at timestep k
$(\cdot)_{k_1:k_2}$	The set of values of a quantity from timestep k_1 to timestep k_2, inclusive
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	A vectrix representing a reference frame in three dimensions
$\underrightarrow{a}$	A vector quantity in three dimensions
$(\cdot)^\times$	The cross-product operator, which produces a skew-symmetric matrix from a 3×1 column
$\mathbf{1}$	The identity matrix
$\mathbf{0}$	The zero matrix
$\mathbb{R}^{M \times N}$	The vectorspace of real $M \times N$ matrices
$\hat{(\cdot)}$	A posterior (estimated) quantity
$\check{(\cdot)}$	A prior quantity

$SO(3)$	The special orthogonal group, a matrix Lie group used to represent rotations
$\mathfrak{so}(3)$	The Lie algebra associated with $SO(3)$
$SE(3)$	The special Euclidean group, a matrix Lie group used to represent poses
$\mathfrak{se}(3)$	The Lie algebra associated with $SE(3)$
$(\cdot)^\wedge$	An operator associated with the Lie algebra for rotations and poses
$(\cdot)^\curlywedge$	An operator associated with the adjoint of an element from the Lie algebra for poses
$\mathrm{Ad}(\cdot)$	An operator producing the adjoint of an element from the Lie group for rotations and poses
$\mathrm{ad}(\cdot)$	An operator producing the adjoint of an element from the Lie algebra for rotations and poses

Chapter 1

Introduction

The unique task of autonomously traversing a human-taught path gives a robot a strong prior on where it is safe to drive (Berczi and Barfoot, 2016). This allows for confident, autonomous navigation through rough, outdoor terrain that would otherwise be inaccessible or require complex, generic, and potentially risky terrain-assessment algorithms. Furthermore, these methods can be implemented to have bounded computation costs and minimal map sizes (Furgale and Barfoot, 2010), making them well suited for long-range navigation. These benefits make autonomous path following appealing for industrial applications that consist of repeated traversals over constrained paths, such as factory floors, orchards, and mines. They are also well-suited to applications that consist of autonomous exploration and retrotraverse such as search-and-rescue and hazardous-exploration robots.

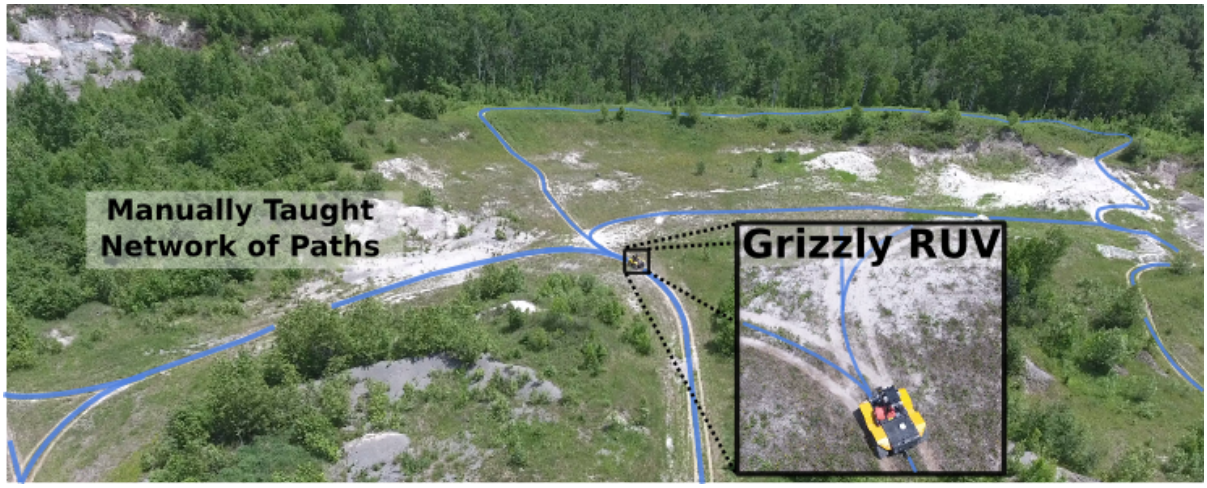


Figure 1.1: A Grizzly Robotic Utility Vehicle (RUV) deployed with an autonomous path-following algorithm navigating a 5 km network of manually taught paths. Applications that rely on repeated traversals of constrained paths will greatly benefit from such algorithms. Examples include mining, agriculture, and patrol robots. Autonomous path-following algorithms that are usable by such applications will need to be able to cope with large-scale maps and appearance change over long periods of time. We are furthermore motivated to develop vision-based algorithms due to the price and ubiquity of passive sensors. Using novel localization and mapping methods developed in this thesis, the robot pictured above autonomously traversed over 140 km over two weeks, experiencing significant appearance changes in the environment (see Section 5.3.2).

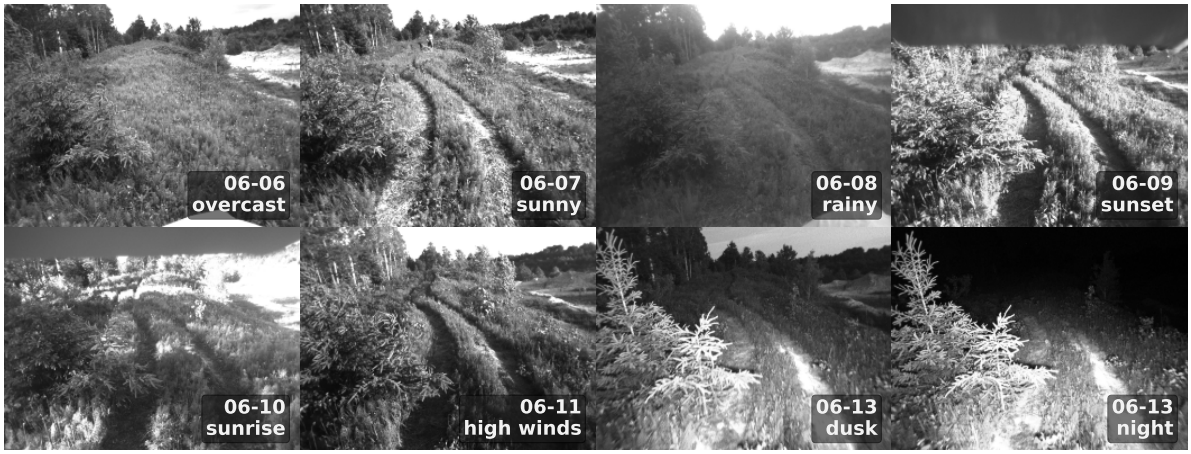
Autonomous path-following systems suited for outdoor applications will require the ability to navigate large-scale environments over long time periods. They will furthermore require constant, metric localization to the manually driven path as the input error signal to a path-tracking controller to ensure minimal drift over time. This thesis presents extensions to the stereo-vision-based autonomous path-following system, Visual Teach & Repeat (VT&R) (Furgale and Barfoot, 2010) that satisfy these requirements. In particular, this thesis focuses on the requirement of navigation over long time periods, which is a non-trivial issue for vision-based systems that operate outdoors. These requirements pose a serious challenge for vision-based systems whose advantages of cost and commercial ubiquity come at the expense of robustness to appearance change. While indoor applications such as those in factories and mines (with suitable lighting) mean appearance change is minimal, outdoor applications such as those in agricultural fields and open-pit mines will require operation in environments with vastly differing appearance. Lighting change is a significant factor over modest time scales, where shadows move throughout the day and cloud cover can change appearance from one minute to the next. Over periods of weeks or months, seasonal changes due to foliage and snow cover can also dramatically affect appearance. Even the robot, through terrain modification from repeated traverses (tire tracks, vegetation damage) can contribute to this appearance change. Examples of these changes can be see in Figure 1.2, which shows the varying appearance of two of our field test sites due to lighting (Figure 1.2a) and winter weather (Figure 1.2b). As a result, the operational domain of path-following systems that rely on vision are typically limited to a few hours outdoors, as highlighted by Furgale and Barfoot (2010), due directly to appearance change.

1.1 Thesis Overview

The structure of this thesis is outlined in Figure 1.3. In Chapter 2, we provide background information for the reader on the baseline autonomous path-following system upon which this thesis builds, Visual Teach & Repeat (VT&R) 1.0 (Furgale and Barfoot (2010)), as well as the mathematical machinery used throughout this thesis to estimate the state of vehicles operating in three dimensions.

In Chapter 3, we present Multi-Channel Localization (MCL): a generic localization and mapping framework used throughout this thesis that provides singular state estimates through the use of multiple channels of information. In the context of MCL, a channel is defined as a stream of measurements usable in a state estimation problem. Consider a system based on sparse visual features with depth, a channel in this case would be any stream of information capable of generating these features. Examples of channels usable by this system include stereo images, RGB-D images, and Light Detection and Ranging (LiDAR)-based intensity images. This multi-channel localization system is used to demonstrate an increase in robustness against appearance change through two novel localization systems.

First, we present a lighting-resistant system that provides metric localization across short-term appearance change due to lighting. Through the use of the multi-channel localization framework, we provide a many-to-one localizer that incorporates data from grayscale images and color-constant images, whose appearance remains constant in the face of changing illumination. We furthermore present background information on color-constant theory and provide a method to experimentally tune the transformation from RGB to color-constant with respect to a given environment. Next, we present a multi-stereo localizer whose robustness to appearance change is increased by extending the field of view of the robot. Though the multi-channel localization system, we fuse data correspondences from separate stereo cam-



(a) Examples of appearance change primarily due to lighting observed over eight days while autonomously following the network of paths displayed in Figure 5.1 during the field test described in Section 5.3.2.

(b) Examples of appearance change primarily due to winter weather observed over four months while autonomously following a path at the University of Toronto during the field test described in Section 5.3.4.

Figure 1.2: Appearance change due to lighting (Figure 1.2a) and winter weather (Figure 1.2b). In both examples, all images were successfully localized metrically with respect to the privileged map image (top left images) using the MEL algorithm detailed in Chapter 4.

eras and increase our system's robustness to appearance change by doubling the amount of inlier feature matches observed by our system. Lastly, we combine these two localizers to achieve multi-stereo, lighting-resistant localization. We demonstrate significantly improved performance over single-camera methods, especially in winter environments that are overwhelmingly difficult for vision-based systems.

These multi-channel localizers are then integrated into the full VT&R 1.0 system, with extensive field results in challenging, outdoor conditions. In the first field test, we demonstrate the lighting-resistant localization system's ability to operate across significant lighting changes. This test consists of manually demonstrating a 1 km loop at the Canadian Space Agency (CSA) in Montreal, Canada, spanning a variety of environments including rocky desert, open grassland, and tall trees. Over the course of four days, the robot autonomously traversed the loop 26 times with an autonomy rate of 99.9%, experiencing nearly every daytime lighting condition. In the next two field tests, the multi-stereo, lighting-resistant system is validated by autonomously traversing two small loops in harsh winter conditions, including a meadow with melting snow and a meadow with full snow cover. We show that the addition of a

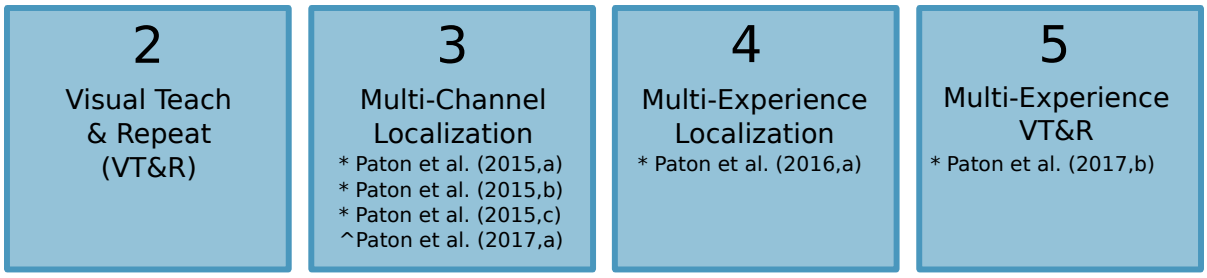


Figure 1.3: Chapter descriptions with relevant publications listed (* conference, ^ journal). Chapters with no listed publications serve as background information for the reader.

second stereo camera to the system significantly increases localization performance compared to the single-camera lighting-resistant method. Despite an autonomy rate of 100%, we show that there is a significant decrease in localization performance when compared to localization in the summer. We provide a detailed analysis on the reasons behind these failures as well as failure points of vision-based localization in these difficult conditions. We conclude that winter environments are a severe limitation on systems that rely on localizing a live view to a static map. Primary reasons include accelerated lighting change from the low elevation of the sun, intense glare from the snow, and rapid appearance change from snow accumulation and melt.

Novel contributions in this chapter include the MCL framework, which provides many-to-one localization from multiple information sources, a MCL-based, lighting-resistant localizer which uses experimentally tuned color-constant images, a MCL-based, multi-stereo localizer that increases the system's field of view, and 26 km of vision-in-the-loop field tests. We also provide insights into the expected performance and limitations of these systems through detailed analysis of multiple field tests in harsh, winter conditions. The extensions to the VT&R 1.0 system presented in this chapter improved localization performance and allowed autonomous path following over multiple days despite significant changes in the lighting of the scene. However, because the system relies on associating appearance between the live experience and a static map captured during manual demonstration of the path, it is limited to an operational window of multiple days in nominal conditions and only a few hours in less ideal conditions such as winter. This is unacceptable for most industrial applications that will need reliable navigation over the time period of months and years. In the next two chapters of the thesis, we address this limitation through a novel, multi-experience localization and mapping framework.

In Chapter 4, we present the primary contribution of this thesis, Multi-Experience Localization (MEL): a metric localization algorithm designed specifically for long-term autonomous path following. While long-term localization techniques exist that make use of multiple experiences, they either provide topological-only localization, require several manually taught experiences in different conditions, or require extensive offline mapping to produce metric localization. For real-world use, we would like a path-following system capable of continuous operation immediately after manual path creation. The MEL algorithm satisfies these requirements and addresses the limitation of appearance change with a single overarching enhancement: the ability to continuously estimate, with uncertainty, the localization between the live experience and a privileged (manual) experience, by using several other intermediate experiences simultaneously to bridge the appearance gap (see Figure 4.1). Our work differs from other systems in that we would like to have only a single manually taught experience (the privileged experience) and add the bridging experiences *during* autonomous operations.

To demonstrate the capability of the MEL algorithm, we conducted three unique experiments. The

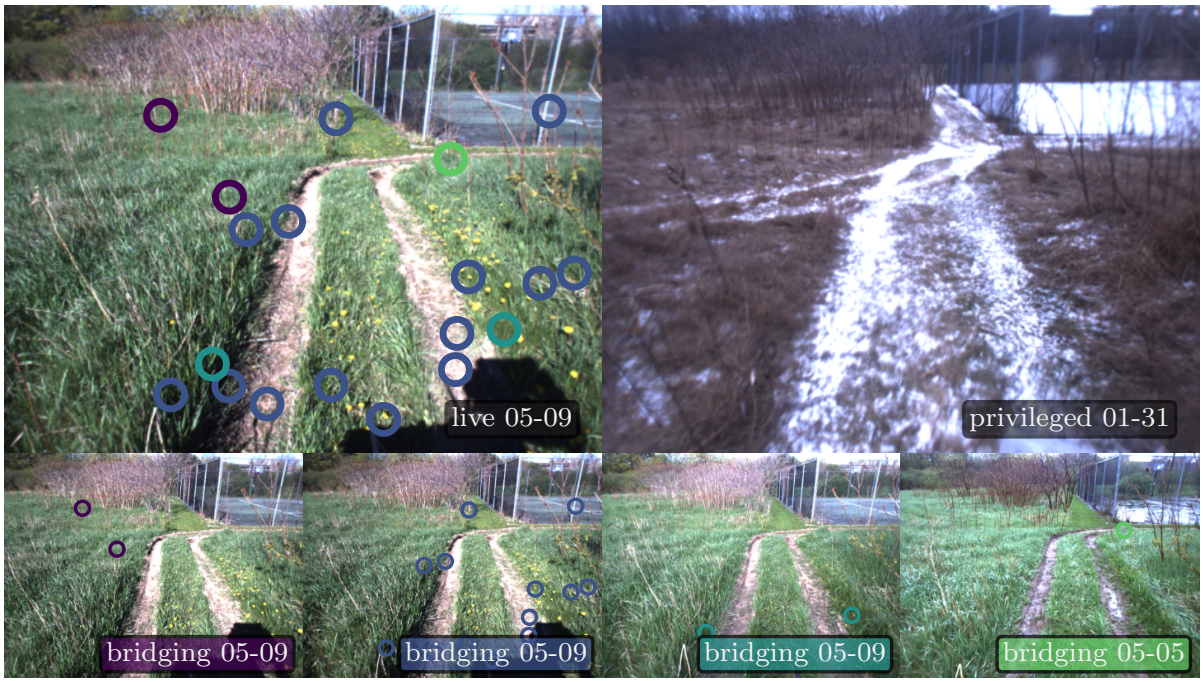


Figure 1.4: Illustration of the Multi-Experience Localization (MEL) algorithm. This metric localization algorithm designed specifically for autonomous path following estimates the pose of a live experience with respect to a manually driven privileged experience with uncertainty using intermediate experiences to *bridge the appearance gap*. Above, the live experience (top left), captured on 05-09, has an inadequate number of feature matches (circles) to the privileged experience (top right), captured on 01-31 to perform localization. With MEL, matches from bridging experiences (bottom row) whose metric positions to the privileged experience are known, are used to localize the live experience robustly with respect to the privileged experience despite extreme seasonal appearance change.

first experiment demonstrates the core concepts of the MEL algorithm: i) data associations between the live view and multiple bridging experiences can be used simultaneously to solve a single state estimation problem, and ii) the appearance gap between the live view and privileged view can be sufficiently bridged in real time using a fixed subset of bridging experiences. This offline performance analysis is conducted on a 9 km subset of the challenging 26 km CSA dataset detailed in Chapter 3, which exhibits significant appearance change due to lighting variation. The second experiment compares the performance of the MEL algorithm to its most related work, Experience-Based Navigation (EBN) (Churchill and Newman, 2013) on a challenging winter data set containing extreme appearance change due to snow fall and melt. The third experiment addresses concerns of the MEL algorithm's ability to minimize localization drift as the appearance of the scene changes. This "photocopy-of-a-photocopy" issue arises from the fact that the MEL algorithm will at some point only match to landmarks from bridging experiences, whose metric relation to the privileged experience is computed from previous, uncertain transformations. This online experiment consists of teaching a small, 50 m straight-line path and autonomously repeating the path back and forth over 180 times while correcting ground truth data with a Leica TotalStation. We show that even after 180 autonomous traversals, the accuracy of the VT&R 2.0 system with respect to the original path remains in the centimeter level.

To summarize, this chapter presents an algorithm capable of long-term metric localization suitable for autonomous path-following algorithms. The most significant novel contribution of this thesis is presented in this chapter: the concept of providing metric localization between a live view and a single privileged experience using bridging experiences gathered during autonomous operation. The novel contributions of this chapter are: i) a data structure that relates multiple experiences together metrically,

ii) a methodology to metrically localize a live experience to a privileged, manually driven experience using several intermediate experiences gathered during autonomous operation, iii) a methodology to bookkeep uncertainties in the multi-experience localizer, accounting for uncertain map landmarks originating from multiple experiences, and iv) experimental evaluations of the MEL system to validate the core ideas of metric localization using many experiences.

In Chapter 5, both the MCL and MEL localization algorithms presented in the previous chapters are integrated into the multi-experience, multi-channel autonomous path-following system, Visual Teach & Repeat (VT&R) 2.0. This vision-based path-following system is capable of safe, long-term navigation over large-scale networks of connected paths in unstructured, outdoor environments. The system makes use of the MCL framework to provide lighting-resistant localization using color-constant images and the MEL algorithm to provide long-term operation capable of localization across seasons. In addition to the MEL algorithm for metric localization, the VT&R 2.0 system also makes use of a suite of multi-experience navigation algorithms whose novel contributions are the work of fellow researchers to provide scalable and safe navigation. In particular, research being conducted in parallel with this thesis provides computational bounds to the MEL algorithm by recommending a fixed subset of experiences to localize against at the start of each state estimation problem. The algorithms developed to recommend experiences in VT&R 2.0 are not novel contributions of this thesis and are only mentioned to provide a complete description of the VT&R 2.0 system.

The VT&R 2.0 system is experimentally validated through three distinct outdoor field tests accumulating over 185 km of vision-in-the-loop autonomous driving across appearance change as dramatic as night vs. day and winter vs spring. In the first field trial, we demonstrate the VT&R 2.0 system's ability to perform online, vision-in-the-loop autonomous path following as the number of total experiences increases. This consisted of manually demonstrating a small 250 m loop at the University of Toronto Institute for Aerospace Studies (UTIAS) on a sunny day in mid summer, 2016. Using the MEL algorithm in conjunction with an appearance-based experience selector, the VT&R 2.0 system was able to continuously repeat the loop 30 times over a period of 30 hours. During the field trial, the MEL algorithm was able to bridge the appearance change from sunset to night with on-board headlights while localizing to the privileged experience taught during the day. This field test demonstrated the MEL algorithm's ability to reliably localize to the privileged experience using a small subset of the total experiences in the map. The second field test was designed to stress test the VT&R 2.0 system's ability to perform autonomous path following on a large-scale network of paths over a longer time period in more difficult, unstructured environments. This consisted of an eleven-day field test in an untended gravel pit in Sudbury, Canada, where we incrementally built and autonomously traversed a 5 km network of paths. Over the span of the field test, the robot logged over 140 km of vision-in-the-loop autonomous driving with an autonomy rate of 99.6% despite experiencing significant appearance change due to lighting and weather, including driving at night using headlights.

The final VT&R 2.0 field test was designed to stress test the system's ability to autonomously follow paths across multiple season's. This field test consisted of repeated autonomous traversals of a small 160 m loop taught in an open meadow at UTIAS. Using the VT&R 2.0 system, the robot autonomously traversed the loop over 160 times over the span of four months experiencing dramatic appearance change due to winter weather, lighting, driving at night, and spring vegetation growth (see figure Figure 1.2b). Despite these challenging appearance changes, the VT&R 2.0 system was able to autonomously repeat the loop with a 99.99% autonomy rate of distance traveled. This chapter provides detailed performance

analysis on each field test with respect to the MEL algorithm and concludes with a discussion on conditions that remain difficult for vision-based navigation. The novel contributions in this chapter are: i) a vision-in-the-loop autonomous path-following system that makes use of a multi-experience localization and mapping framework to provide inter-seasonal autonomy and nighttime autonomy with on-board headlights, and ii) extensive long-term field tests of the system involving autonomy in unstructured, outdoor environments with rapidly changing appearance, covering over 178 km of vision-in-the loop autonomous driving.

The final chapter summarizes the novel contributions of the thesis and provides a discussion on current limitations of the VT&R 2.0 system and future work.

Chapter 2

Visual Teach & Repeat

This chapter serves as background information on the short-term autonomous path following system that this thesis extends upon, Visual Teach & Repeat (VT&R) 1.0. This chapter also provides a primer on the state estimation machinery used in all of the autonomous path following systems presented in this thesis as well as a system-level overview of VT&R 1.0. As this is a background chapter, there are no novel contributions presented.

2.1 State Estimation Primer

In this section, we provide background information on the state estimation machinery used in the autonomous path following systems presented in this thesis. For detailed derivations in the mathematical formulas presented here, we refer the readers to Barfoot (2017).

2.1.1 Three-Dimensional Geometry

This section provides information on how to represent *rotations* and *poses* in three dimensions. For derivations and information on how to perturb and linearize rotations and poses we refer the reader to Chapter 6 of Barfoot (2017).

Matrix Lie Groups

Poses and rotation in this thesis are represented by matrix Lie groups. A *group* is defined as a set of elements with an operation that combines any two of its elements into a third element that remains in the same set. A Lie group is a group with the property that the group operations are *smooth*, or rather differential calculus can be used. A matrix lie group is simply a group where the elements of the group are matrices.

Rotations Rotations in this thesis are represented with the *special orthogonal* group, $SO(3)$, which can be defined as the set of valid rotation matrices:

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3\times 3} \mid \mathbf{C}\mathbf{C}^T = \mathbf{1}, \det \mathbf{C} = 1 \right\} \tag{2.1}$$

The condition, $\mathbf{C}\mathbf{C}^T = \mathbf{1}$, imposes six constraints on the nine-parameter matrix, $\mathbf{C}$, reducing the number of degrees of freedom to three. The second constraint ensures that $\mathbf{C}$ is a *proper* rotation.

Poses Poses, which consist of a translation and a rotation, in this thesis are represented with the *special Euclidean* group, $SE(3)$. $SE(3)$ poses are defined as the set of valid transformation matricies:

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \;\middle|\; \mathbf{C} \in SO(3),\, \mathbf{r} \in \mathbb{R}^3 \right\}. \tag{2.2}$$

Lie Algebra

Every matrix Lie group is associated with a Lie algebra. A Lie algebra consists of a vectorspace, $\mathbb{V}$, over some field, together with a binary operation called the Lie backet, $[\cdot,\cdot]$, that satisfies the following four properties: i) *closure*, ii) *bilinearity*, iii) *alternating*, and iv) *Jacobi identity*.

Rotations The Lie algebra associated with the $SO(3)$ Lie group is given by:

$$\begin{aligned} \text{vectorspace:} \quad & \mathfrak{so}(3) = \left\{ \boldsymbol{\Phi} = \boldsymbol{\phi}^\wedge \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\phi} \in \mathbb{R}^3 \right\} \\ \text{field:} \quad & \mathbb{R}, \\ \text{Lie bracket:} \quad & [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2] = \boldsymbol{\Phi}_1\boldsymbol{\Phi}_2 - \boldsymbol{\Phi}_2\boldsymbol{\Phi}_1, \end{aligned}$$

where

$$\boldsymbol{\phi}^\wedge = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}, \quad \boldsymbol{\phi} \in \mathbb{R}^3. \tag{2.3}$$

Poses The Lie algebra associated with the $SE(3)$ Lie group is given by:

$$\begin{aligned} \text{vectorspace:} \quad & \mathfrak{se}(3) = \left\{ \boldsymbol{\Xi} = \boldsymbol{\xi}^\wedge \in \mathbb{R}^{4\times 4} \mid \boldsymbol{\xi} \in \mathbb{R}^6 \right\} \\ \text{field:} \quad & \mathbb{R} \\ \text{Lie bracket:} \quad & [\boldsymbol{\Xi}_1, \boldsymbol{\Xi}_2] = \boldsymbol{\Xi}_1\boldsymbol{\Xi}_2 - \boldsymbol{\Xi}_2\boldsymbol{\Xi}_1, \end{aligned}$$

where

$$\boldsymbol{\xi}^\wedge = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4},\ \boldsymbol{\rho}, \boldsymbol{\phi} \in \mathbb{R}^3 \tag{2.4}$$

Exponential Map

The *exponential map* can be used to convert between Lie groups and Lie algebras. The matrix exponential and logarithm are given by:

$$\exp(\mathbf{A}) = \mathbf{1} + \mathbf{A} + \frac{1}{2!}\mathbf{A}^2 + \frac{1}{3!}\mathbf{A}^3 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!}\mathbf{A}^n, \tag{2.5}$$

$$\ln(\mathbf{A}) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} (\mathbf{A} - \mathbf{1})^n \tag{2.6}$$

respectively, where $\mathbf{A} \in \mathbb{R}^{M\times M}$.

Rotations Elements in the matrix Lie group, $SO(3)$, and Lie algebra, $\mathfrak{so}(3)$, can be related through the exponential map as follows:

$$\begin{aligned}\mathbf{C} &= \exp(\boldsymbol{\phi}^\wedge) && (2.7)\\ &= \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\phi}^\wedge)^n && (2.8)\\ &= \cos(\phi)\mathbf{1} + (1-\cos(\phi))\,\mathbf{a}\mathbf{a}^T + \sin\phi\mathbf{a}^\wedge, && (2.9)\end{aligned}$$

where $\boldsymbol{\phi} = \{\mathbf{a}, \phi\}$ is the axis-angle representation of a rotation matrix. Conversion from $\mathfrak{so}(3)$ to $SO(3)$ can also be achieved through the logarithmic map:

$$\boldsymbol{\phi} = \ln(\mathbf{C})^\vee, \tag{2.10}$$

although not uniquely. The operator, $(\cdot)^\vee$, is the inverse of (2.3).

Poses Elements in the matrix Lie group, $SE(3)$, and Lie algebra, $\mathfrak{se}(3)$, can be related through the exponential map as follows:

$$\begin{aligned}\mathbf{T} &= \exp(\boldsymbol{\xi}^\wedge) && (2.11)\\ &= \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\xi}^\wedge)^n && (2.12)\\ &= \sum_{n=0}^{\infty} \frac{1}{n!}\left(\begin{bmatrix}\boldsymbol{\rho}\\ \boldsymbol{\phi}\end{bmatrix}^\wedge\right)^n && (2.13)\\ &= \sum_{n=0}^{\infty} \frac{1}{n!}\left(\begin{bmatrix}\boldsymbol{\phi}^\wedge & \boldsymbol{\rho}\\ \mathbf{0}^T & 1\end{bmatrix}\right)^n && (2.14)\\ &= \begin{bmatrix}\sum_{n=0}^{\infty} \frac{1}{n!}\left(\boldsymbol{\phi}^\wedge\right)^n & \left(\sum_{n=0}^{\infty} \frac{1}{(n+1)!}\left(\boldsymbol{\phi}^\wedge\right)^n\right)\boldsymbol{\rho}\\ \mathbf{0}^T & 1\end{bmatrix} && (2.15)\\ &= \begin{bmatrix}\mathbf{C} & \mathbf{r}\\ \mathbf{0}^T & 1\end{bmatrix} \in SE(3), && (2.16)\end{aligned}$$

where

$$\mathbf{r} = \mathbf{J}\boldsymbol{\rho} \in \mathbb{R}^3,\ \mathbf{J} = \sum_{n=0}^{\infty} \frac{1}{(n+1)!}\left(\boldsymbol{\phi}^\wedge\right)^n, \tag{2.17}$$

and the matrix $\mathbf{J}$ is the *left jacobian* of $SO(3)$.

We can also go the other direction, but not uniquely, from $\mathfrak{se}(3)$ to $SE(3)$ through the *logarithmic* map as follows:

$$\boldsymbol{\xi} = \ln(\mathbf{T})^\vee = \ln(\left(\begin{bmatrix}\mathbf{C} & \mathbf{r}\\ \mathbf{0}^T & 1\end{bmatrix}\right)^\vee = \begin{bmatrix}\mathbf{J}^{-1}\mathbf{r}\\ \boldsymbol{\phi}\end{bmatrix}, \tag{2.18}$$

where $\mathbf{J}^{-1}$ is the inverse jacboian of $SO(3)$.

2.1.2 Nonlinear Estimation

In the autonomous path following systems detailed in this thesis, the $SE(3)$ state of the robot is obtained through discrete-time, batch, nonlinear state estimation. We provide a high-level overview of the machinery used in our systems to estimate the motion of the robot as well as the robot's position with respect to the manually driven path. Further information on nonlinear estimation can be found in Chapter 4 of Barfoot (2017).

Problem Setup We define the following nonlinear motion and observation models used in these estimation problems as:

$$\text{motion model:} \quad \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{w}_k\right), \tag{2.19}$$

$$\text{observation model:} \quad \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{n}_k), \tag{2.20}$$

where k is the discrete-time index and K is its maximum. The function, $\mathbf{f}(\cdot)$, is the nonlinear motion model and the function $\mathbf{g}(\cdot)$ is the nonlinear observation model. The variables in the models take the following form:

$$\text{system state:} \quad \mathbf{x}_k \in \mathbb{R}^N \tag{2.21}$$

$$\text{initial state:} \quad \mathbf{x}_0 \in \mathbb{R}^N \sim \mathcal{N}\left(\check{\mathbf{x}}_0, \check{\mathbf{P}}_0\right) \tag{2.22}$$

$$\text{input:} \quad \mathbf{v}_k \in \mathbb{R}^N \tag{2.23}$$

$$\text{process noise:} \quad \mathbf{w}_k \in \mathbb{R}^N \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_k\right) \tag{2.24}$$

$$\text{measurement:} \quad \mathbf{y}_k \in \mathbb{R}^M \tag{2.25}$$

$$\text{measurement noise:} \quad \mathbf{n}_k \in \mathbb{R}^M \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_k\right) \tag{2.26}$$

$$\tag{2.27}$$

The noise variables and initial state knowledge are assumed to be independent of each other and of themselves at different timestamps.

Maximum A Posteriori (MAP) This method finds the single best estimate of the posterior, $\hat{\mathbf{x}}$, of the state of the system, $\mathbf{x} = \mathbf{x}_{0:K}$, given the prior information, $\mathbf{v} = (\mathbf{x}_0, \mathbf{v}_{1:K})$, and measurements, $\mathbf{y} = \mathbf{y}_{0:K}$ at *all* timesteps, $\{0..k\}$. This is defined as Maximum A Posteriori (MAP) and is formulated as follows:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{v}, \mathbf{y}). \tag{2.28}$$

Using Baye's rule, Equation 2.28 can be rewritten as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{v}, \mathbf{y}) = \arg\max_{\mathbf{x}} \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{v})p(\mathbf{x}|\mathbf{v})}{p(\mathbf{y}|\mathbf{v})} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\mathbf{v}), \tag{2.29}$$

where the denominator has been dropped because it does not depend on $\mathbf{x}$ and the prior term, $\mathbf{v}$, is dropped in $p(\mathbf{y}|\mathbf{x}, \mathbf{v})$ because it has no effect on the measurements, $\mathbf{y}$.

Because we are assuming that all of the noise variables are independent of each other, the terms in

Equation 2.29 can be rewritten as:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{k=0}^{K} p(\mathbf{y}_k \,|\, \mathbf{x}_k), \tag{2.30}$$

$$p(\mathbf{x}|\mathbf{v}) = p(\mathbf{x}_0 \,|\, \check{\mathbf{x}}_0) \prod_{k=1}^{K} p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{v}_k). \tag{2.31}$$

To make the optimization process easier, the logarithm of both sides is taken:

$$\ln(p(\mathbf{y}|\mathbf{x})p(\mathbf{x}\,|\mathbf{v})) = \ln p(\mathbf{x}_0 \,|\, \check{\mathbf{x}}_0) + \sum_{k=1}^{K} \ln p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{v}_k) + \sum_{k=0}^{K} \ln p(\mathbf{y}_k \,|\, \mathbf{x}_k). \tag{2.32}$$

This reformulation does not affect our optimization problem as the logarithm function is monotonically increasing. If the noise variables are assumed to be normally distributed with mean values of $\mathbf{0}$, then the components of Equation 2.32 can be defined as:

$$\begin{aligned} \ln p(\mathbf{x}_0 \,|\, \check{\mathbf{x}}_0) = &-\frac{1}{2} \left(\mathbf{x}_0 - \check{\mathbf{x}}_0\right)^T \check{\mathbf{P}}_0^{-1} \left(\mathbf{x}_0 - \check{\mathbf{x}}_0\right) \\ &- \underbrace{\frac{1}{2} \ln \left((2\pi)^N \det \check{\mathbf{P}}_0\right)}_{\text{independent of } \mathbf{x}}, \end{aligned} \tag{2.33}$$

$$\begin{aligned} \ln p(\mathbf{x}_k \,|\, \mathbf{x}_{k-1}, \mathbf{v}_k) = &-\frac{1}{2} \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k\right)^T \mathbf{Q}_k^{-1} \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k\right) \\ &- \underbrace{\frac{1}{2} \ln \left((2\pi)^N \det \mathbf{Q}_k\right)}_{\text{independent of } \mathbf{x}}, \end{aligned} \tag{2.34}$$

$$\begin{aligned} \ln p(\mathbf{y}_k|\mathbf{x}_k) = &-\frac{1}{2} \left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{0})\right)^T \mathbf{Q}_k^{-1} \left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{0})\right) \\ &- \underbrace{\frac{1}{2} \ln \left((2\pi)^N \det \mathbf{R}_k\right)}_{\text{independent of } \mathbf{x}}. \end{aligned} \tag{2.35}$$

We then define the following functions, removing terms that do not contain the state variable, $\mathbf{x}$, which are seeking to optimize:

$$\begin{aligned} J_{v,k}(\mathbf{x}) &= \begin{cases} \frac{1}{2} \left(\mathbf{x}_0 - \check{\mathbf{x}}_0\right)^T \check{\mathbf{P}}_0^{-1} \left(\mathbf{x}_0 - \check{\mathbf{x}}_0\right), & k = 0 \\ \frac{1}{2} \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k\right)^T \mathbf{Q}_k^{-1} \left(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k, \mathbf{0}) - \mathbf{x}_k\right), & k = 1 \dots K \end{cases}, \\ J_{y,k}(\mathbf{x}) &= \frac{1}{2} \left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{0})\right)^T \mathbf{Q}_k^{-1} \left(\mathbf{y}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{0})\right), \qquad k = 0 \dots K, \end{aligned}$$

which are all squared Mahalanobis distances. With this new formulation, the MAP problem can be redefined as minimizing the following objective function with respect to the state variable, $\mathbf{x}$:

$$J(\mathbf{x}) = \sum_{k=0}^{K} \left(J_{v,k}(\mathbf{x}) + J_{y,k}(\mathbf{x})\right), \tag{2.36}$$

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} J(\mathbf{x}), \tag{2.37}$$

which finds the best estimate, $\hat{\mathbf{x}}$, in order to maximize the joint likelihood of all the data we have.

To find the minimum of (2.36), there are many nonlinear optimization techniques that can be employed. Typical techniques include Newton's Method, which iteratively approximates the differentiable objective function by a quadratic function and jumps to the minimum and the Gauss-Newton method which makes approximations to the Newton's method to improve speed and simplify calculations.

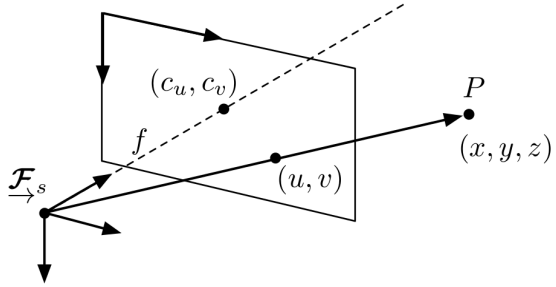
2.1.3 Stereo Geometry

This section briefly overviews the stereo camera model used in this thesis to project 3D points in world coordinates into stereo camera observations in pixel coordinates. Passive cameras are appealing for robotics as they are commercially ubiquitous, inexpensive, and can infer motion of a vehicle and the shape of the world. An illustration of the basic frontal projection camera model, commonly used in computer vision, can be seen in Figure 2.1. Using this model a point, $\boldsymbol{\rho} = (x, y, z)$, in the sensor coordinate frame, $\underrightarrow{\mathcal{F}}_s$ can be projected onto the image sensor plane as a sensor observation with pixel coordinates, (u, v) through the following mapping:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{s}(\boldsymbol{\rho}) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{\mathbf{P}} \underbrace{\begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{2.38}$$

where P is a projection matrix to remove the bottom row from the homogeneous point representation and K is the *intrinsic parameter matrix* of the sensor, containing the focal length, (f_u, f_v), in pixels and the offset of the image origin from the optical axis intersection, (c_u, c_v). This mapping shows that with only one perspective camera the depth of the point, $\boldsymbol{\rho}$, is not recoverable, and only the shape of the metric structure can be obtained with multiple views of the same scene. With a single perspective camera, depth can be recovered if assumptions are made about the structure of the world of if an additional sensor can provide scale such as a GPS system or an IMU.

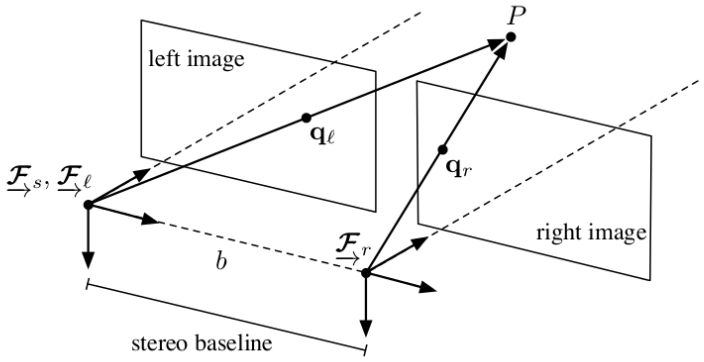
Stereo vision simplifies this problem by providing scale to the scene through known extrinsic information between two perspective cameras, which allows for the recovery of the euclidean structure. An example of this is the left-camera stereo model, depicted in Figure 2.2, which is used throughout



Credit: *Barfoot (2017)*

Figure 2.1: Illustration of the frontal projection camera model with intrinsic parameters shown.

this thesis. In this model, two perspective cameras are rigidly aligned along the x-axis with a known translation, b, called the stereo baseline. Using this model a homogeneous point, $\boldsymbol{\rho} = (x, y, z, 1)$, in the left camera coordinate frame, $\underrightarrow{\mathcal{F}}_l$ can be projected onto the left and right image sensor plane as an



Credit: *Barfoot (2017)*

Figure 2.2: Illustration of the left-stereo camera model consisting of a left and right camera with respective coordinate frames, F_l, and F_r. Both cameras are pointed in the same direction and aligned on the same horizontal axis, with a known translation b. A point, $\mathbf{P}$, observed by both cameras is represented in the coordinate frame of F_s. In the case of the left camera model, F_s and F_l are identical.

observation with pixel coordinates, (u_l, v_l, u_r, v_r), through the following mapping:

$$\begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = \mathbf{g}(\boldsymbol{\rho}) = \underbrace{\begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ f_u & 0 & c_u & -f_u b \\ 0 & f_v & c_v & 0 \end{bmatrix}}_{\mathbf{M}} \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.39}$$

where we assume that the left and right camera have the same intrinsic properties.

2.2 Sparse Stereo Visual Odometry (VO)

The autonomous path following systems presented in this paper are all built upon the state estimation machinery provided by stereo Visual Odometry (VO)—the estimation of a robot's motion using stereo cameras. This section provides an overview of the main components that make up a sparse stereo VO pipeline. Sparse VO is based on the detection and tracking of salient keypoints between images. The core components of the sparse VO pipeline were introduced by Moravec in his PhD thesis (Moravec, 1980), they are: i) keypoint extraction, ii) left-right matching, iii) keypoint tracking, iv) outlier rejection, and v) nonlinear refinement. Sparse VO was adapted specifically for stereo cameras by Matthies in his PhD thesis (Matthies, 1989), and later integrated and successfully deployed on the Jet Propulsion Laboratory (JPL)'s Mars Exploration Rover (MER) platforms, Spirit and Opportunity (Maimone et al., 2007), and the Mars Science Laboratory (MSL) Curiosity robot (Johnson et al., 2008). To date, the Opportunity and Curiosity robots are continuing to explore the surface of Mars. Since then, sparse stereo VO has been heavily researched with improvements to every aspect of the algorithm. This section provides overviews and background information on the major components of the sparse stereo VO pipeline, using the 3D state estimation machinery provided in the previous sections.

2.2.1 Pipeline Overview

The sparse VO pipeline is shown in Figure 2.3, the input to the system is a rectified stereo image pair, and the output is an estimation of the relative $SE(3)$ transformation between the input image pair and a previous image pair.

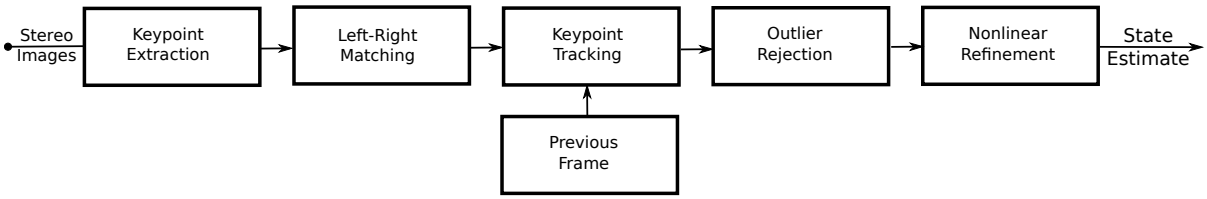


Figure 2.3: High-level stereo VO pipeline for systems that rely on a visual-feature front end with landmark-based detection and matching, and bundle adjustment state estimation. The input to the system is a pair of rectified stereo images. The output is a 6DOF estimate of the state of the robot in the camera frame. The steps of the pipeline are as follows: i) Visual features with keypoints and descriptors are extracted from the pair of stereo images, ii) Keypoints are matched between left and right, and are triangulated to obtain a depth for each feature, iii) Keypoints from a previous frame are tracked in the live frame using a matching scheme, iv) An algorithm is run to reject outliers and obtain an initial estimate, and v) An estimate of the state is obtained through a nonlinear optimization algorithm.

Keypoint Extraction Assuming a pair of rectified, undistorted stereo images as input, the first stage of sparse stereo VO is visual feature extraction. Visual features are points in an image that are interesting, repeatable, and typically invariant to viewpoint change and to a lesser degree, lighting. Extraction begins by detecting visual features to obtain a list of keypoints for both left and right images. A keypoint encodes the pixel coordinate of the center of a visual feature as well as information such as scale and measurement uncertainty. Examples of keypoint detection methods include searching for corners (Harris and Stephens, 1988), blobs of light against dark backgrounds and vice-versa (Lowe, 2004; Agrawal et al., 2008; Bay et al., 2008), and high scoring results from binary pixel comparisons (Calonder et al., 2010; Leutenegger et al., 2011; Rublee et al., 2011). Upon completion of keypoint detection in both the left and right images, a visual descriptor for each keypoint is calculated. Descriptors are data products that are used to match keypoints between images and are typically invariant to scale, and often rotation and changes in the global intensity of the image. In the context of stereo VO, keypoint matching is used to find data correspondences between left and right images to triangulate a 3D stereo measurement, as well as track keypoints across stereo image sequences. The end result of feature extraction is a set of visual features for each image containing keypoints, $\{\mathbf{k}_0, \mathbf{k}_1, \ldots, \mathbf{k}_n\}$, and visual descriptors, $\{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n\}$, where the keypoint i is of the form:

$$\mathbf{k}_i = \bar{\mathbf{k}}_i + \delta\mathbf{k}_i, \quad \delta\mathbf{k}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_i) \tag{2.40}$$

where $\bar{\mathbf{k}}_i$ is the 4×1 mean vector containing the left and right keypoint information, $\mathbf{k}_i = (u_l, v_l, u, v_r)$ and $\mathbf{R}_i$ is the 4×4 covariance of the measurement.

Left-Right Matching This stage of the pipeline finds data correspondences between an input pair of visual features extracted from the left and right stereo images. Given a visual feature from the left

image, a corresponding feature can be found in the right image by comparing keypoints that lie along the epipolar line of the stereo camera. Because we make use of the left stereo-camera model described in Section 2.1.3, the epipolar line aligns with the pixel rows of the left-right pair, reducing the search space to keypoints with similar vertical pixel coordinates. After matching, the result is a collection of stereo measurements $\{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_n\}$, and visual descriptors, $\{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n\}$. Where the stereo measurement i is of the form:

$$\mathbf{y} = \begin{bmatrix} \bar{\mathbf{k}}_l \\ \bar{\mathbf{k}}_r \end{bmatrix} + \delta\mathbf{y}, \quad \delta\mathbf{y} = \begin{bmatrix} \mathbf{R}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_r \end{bmatrix}, \tag{2.41}$$

where $\bar{\mathbf{k}}_l, \bar{\mathbf{k}}_r$ is the left and right mean keypoint measurement, respectively, and $\mathbf{R}_l, \mathbf{R}_r$ is the left and right measurement covariance, respectively. The descriptor associated with the stereo measurement is typically chosen to be the left keypoint descriptor. The final step of left-right matching is to triangulate world coordinates for each stereo measurement, resulting in a list of 3D positions with uncertainty.

Keypoint Tracking In this step, stereo measurements from the input images are matched to visual features established as landmarks in the map. Commonly, the map is simply the set of visual features extracted in the previous stereo pair. One of the more expensive steps of a VO pipeline is associating data between images. This is typically done in either a brute-force manner, using a directed search (projecting images based on an initial guess, etc....), or using a data structure such as a k-d tree. For example, the VO running on the MER robots project features from the map image into the query image using an initial guess from wheel odometry.

Outlier Rejection The keypoint tracking module provides a set of raw feature matches between a map and query image pair. Due to the nature of visual features, these matches typically contain a significant number of outliers. It is therefore common practice to employ an outlier rejection algorithm before using the matches to estimate the state. The VT&R 1.0 system uses the RANdom SAmple Consensus (RANSAC) algorithm introduced by Fischler and Bolles (1981) with a fast, closed-form solution for the 6DOF rotation first established in robotics by Horn (1987) and further refined by Umeyama (1991) to ensure proper rotation matrices. While still fairly commonplace in VO systems, RANSAC has seen many small improvements to stability and speed. These improvements involve enhancements such as local re-optimization, priority ordering of data and pruning of unlikely hypotheses. Recently, these improvements have been formulated in a universal framework under the Universal Framework for RANSAC (USAC) algorithm Raguram et al. (2013).

Apart from RANSAC, the use of robust cost functions can be used to lessen the effect of outliers in the state estimate. These cost functions are designed to down weight the influence of correspondences whose error is large compared to the current hypothesis. A comparison of these cost functions in the context of VO systems is provided in MacTavish and Barfoot (2015). The primary benefit of these functions is outlier rejection with little to no computational overhead.

Nonlinear Refinement After outlier rejection is completed, inlier data correspondences are used to refine the estimate of the motion undergone by the camera using the nonlinear MAP estimation technique described in Section 2.1.2. In stereo VO, this consists of minimizing the sum of squared map landmark reprojection errors weighted by the uncertainty of the live landmark measurements. A reprojection error

consists of taking the difference between the landmark measurement in the live image frame and the map landmark, transformed by the estimated camera motion and reprojected into the live image frame using the stereo camera model.

2.3 Visual Teach & Repeat

The VT&R 1.0 system is the foundation on which this this thesis is built. It enables robots to autonomously repeat large-scale trees of paths previously driven by human operators using only a stereo camera.

2.3.1 System Overview

This section provides an overview of the following components of the VT&R 1.0 system: i) map representation, ii) map construction, iii) autonomous path following, and iv) metric localization.

Map Representation The VT&R 1.0 system represents its map, illustrated in Figure 2.4, as a topometric pose graph. Vertices (black triangles) in the graph, each with a reference frame, $\underrightarrow{\mathcal{F}}$, store raw sensor observations and triangulated 3D landmarks (black stars) with associated descriptors. Landmark positions in the graph are represented in coordinates *relative* to the vertex they are stored in. Edges in the graph (blue, lines) link vertices metrically with a relative $SE(3)$ transformation. These data products can be thought of as the output of the sparse VO pipeline described in Section 2.2. 3D landmarks in the system are represented by triangulated Speeded Up Robust Features (SURF) visual features from grayscale stereo image pairs using the gpusurf library developed by Furgale and Tong (2010).

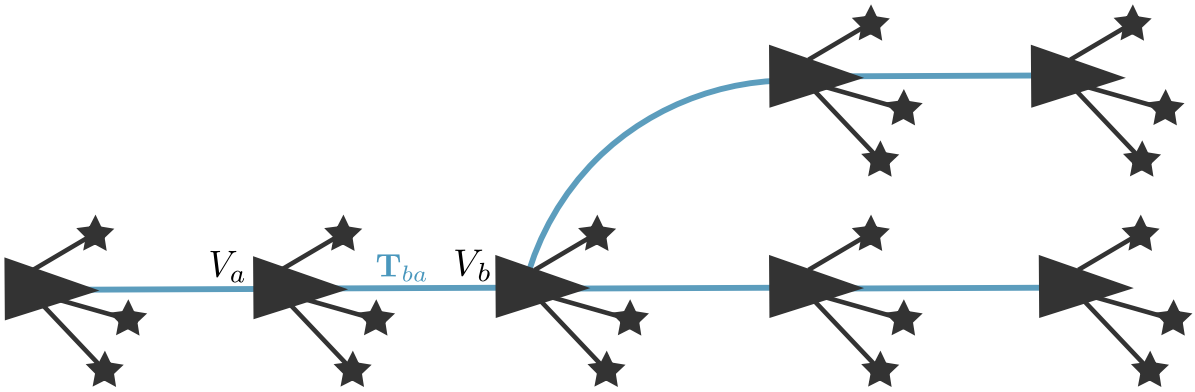


Figure 2.4: Overview of the topometric pose graph used to represent the tree of paths used by the VT&R 1.0 system. Vertices (black triangles) in the graph represent a robot's pose at a key time and contain triangulated stereo landmarks (black stars) with 3D positions and descriptors. Vertices are connected through edges (blue lines) and are related metrically with a relative, $SE(3)$ transformation. This map structure is built during demonstration of the path by a human operator in the teaching phase. Once established, the robot can use this data structure to autonomously follow a route through the graph.

Generic State Estimation In the VT&R 1.0 system, the goal of both localization and VO is to estimate the relative motion of the stereo camera between the current view at time k, $\underrightarrow{\mathcal{F}}_k$, and a reference frame, $\underrightarrow{\mathcal{F}}_m$. This motion can be represented by an $SE(3)$ transformation matrix, $\mathbf{T}_{k,m}$, which

takes points from $\underrightarrow{\mathcal{F}}_m$ into $\underrightarrow{\mathcal{F}}_k$. This is accomplished through the use of the sparse stereo VO pipeline described in Section 2.2. In the case of VO, the reference frame is the previous frame while in the case of localization, the reference frame is a local submap obtained by relaxing a window of vertices centered around the vertex closest to the vehicle. In both cases, we wish to find the estimate of $\mathbf{T}_{k,m}$ that minimizes the reprojection error of all of the landmark observations after they are transformed and reprojected into the image plane. For a given keypoint measurement of landmark j, $\mathbf{y}_{j,k}$, and an observation of the landmark from the reference frame, $\mathbf{p}_m^{j,m}$, the error term, $\mathbf{e}_{j,k}$ is given by

$$\mathbf{e}_{j,k} = \mathbf{y}_{j,k} - \mathbf{g}(\mathbf{T}_{k,m}\mathbf{p}_m^{j,m}), \tag{2.42}$$

where $\mathbf{g}(\cdot)$, is the stereo observation model that transforms points into the image sensor plane. Each keypoint also contains an uncertainty, $\mathbf{Q}_j$, of the measurement of landmark j. The goal of the solver is to minimize the following objective function with respect to the camera transformation, $\mathbf{T}_{k,m}$:

$$J_k = \frac{1}{2}\sum_{j=1}^{n} \mathbf{e}_{j,k}^T \mathbf{Q}_j^{-1} \mathbf{e}_{j,k} + J_{pos}, \tag{2.43}$$

where $(\mathbf{e}_{1,k}, \ldots, \mathbf{e}_{n,k})$ is the set of errors associated with data correspondences from all measured landmarks, and J_{pos} is a prior term on motion. J_{pos} minimizes the error between the posterior transform, $\mathbf{T}_{k,m}$, and a prior transform, $\check{\mathbf{T}}_{k,m}$. In the case of VO, $\check{\mathbf{T}}_{k,m}$ is a no-motion prior, and in the case of localization, $\check{\mathbf{T}}_{k,m}$ is the result of VO. To minimize this objective function, the equation is linearized and then iteratively refined based on the nonlinear MAP techniques described in Section 2.1.2 through the Levenberg-Marqardt algorithm (Levenberg, 1944). The result is a transformation that minimizes the sum of reprojection errors for all measured landmarks to the reference frame. For a thorough review of information on nonlinear optimization we refer the readers to Chapter 4 of Barfoot (2017).

Teach Phase During the teach phase, the robot is manually driven while building the map data structure using the VO pipeline detailed in Section 2.2. This map consists of a topometric pose graph of vertices linked by relative transformations as described in (Furgale and Barfoot, 2010). Vertices in the map are constructed when the robot's motion exceeds a specified threshold, forcing an evenly distributed map. To build a branch from the main path in the map, the user can command the robot to autonomously traverse the path to a desired branching point while in the repeating phase, and then command the robot to begin branching. At this point a localization problem using the generic state estimation pipeline is performed to compute the robot's position with respect to the closest vertex in the map. This computes the first vertex of the branch as well as the relative, $SE(3)$ transformation between the two vertices. After this process is complete the user may drive the robot to begin creating the new branch.

Repeat Phase To autonomously repeat on the manually taught tree of paths, the user first provides a topological estimate of the robot's position on the tree. A localization search is then conducted, starting in the local area surrounding the estimate. Upon successful localization to the tree, the VT&R 1.0 system begins interfacing VO and localization in a predictor/corrector fashion to provide estimates of the robot's position relative to the path. This allows the VT&R 1.0 system to rely on VO estimates in areas where localization is poor. In areas of continual localization failures, the system is allowed to drive

up to 20 m on dead reckoning until the robot stops and the traversal is declared a failure. Localization is achieved by comparing the stream of grayscale stereo data to a local submap pulled from the vertex in the path closest to the robot. This submap is computed from a fixed number of vertices centered at the estimated closest vertex and relaxed into a single coordinate frame. Doing this allows the localization complexity to be constant with respect to the size of the total map. In the case of a localization success, the VO solution is used as a prior, in the case of a localization failure, the VO solution is propagated from the last localization estimate. This information is continuously fed to a path-tracking controller at the frame rate of the sensor to keep the robot on the path. Path tracking is accomplished using Model Predictive Control (Rawlings and Mayne, 2009).

2.3.2 Limitations

The VT&R 1.0 system is effective at providing long-range autonomy with constant-time localization when there is minimal appearance change in the scene. In such an environment, a sufficient number of inlier keypoint matches between the live view and the map can be recovered to provide centimeter-level metric localization to the path-tracking controller. However, the system's reliance on tracking

Figure 2.5: Example of daily appearance change due to lighting on a sunny day. This sequence shows the same scene at different times of day, namely, 10:58, 13:10, 15:17, and 17:44. On sunny days such as this, the operational window of the VT&R 1.0 system is limited to only a few hours.

visual features from grayscale images between a live view and a static map generated during manual demonstration makes it highly susceptible to environments that are affected by lighting change. An example of this appearance change is highlighted in Figure 2.5, which shows the appearance of the same scene across seven hours on a sunny day. In outdoor environments such as this, the number of inlier feature matches to the taught path dramatically decreases as the shadows move across the scene.

If the number of inlier matches drops too low, the system will be forced to rely on VO, and will eventually fail at following the taught path. Figure 3.23 shows an illustration of the trend associated with the number of inlier matches typically observed over the course of a day. This figure sums up the experience collected over the many field tests of the system. On overcast days, there is a gradual decline in keypoint matches, because the appearance of the scene is generally constant. This is not true on sunny days, where an early drop is caused by the sun changing position and creating sharp, moving shadows on the ground. Inlier match counts begins to rise again at the beginning of twilight, when the light from the sun is not directly observable, generating a shadow-less environment similar to an overcast day. This rapid decrease of inlier matches on sunny days make the operational window of the VT&R 1.0 system limited to a few hours on sunny days in outdoor environments.

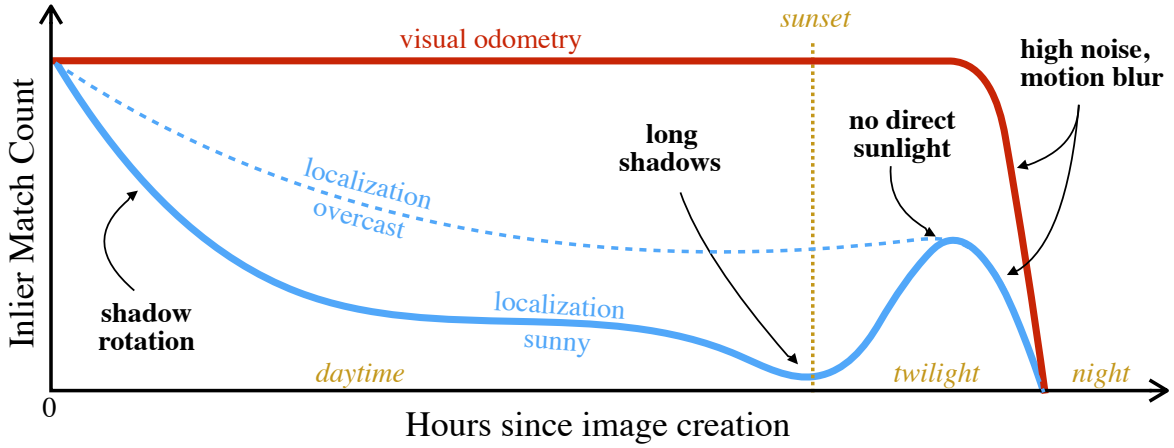


Figure 2.6: Illustration of the evolution of the number of inlier feature matches through a nominal day. Time zero corresponds to when the reference images are collected (teaching phase) and the blue line represent the typical slow degradation of the number of matches when matching current images to the teaching phase. The difference between a sunny day (solid line) and an overcast day (dashed line) is also included. The red line represents the number of features used during **vo!**, which stays constant up to the limit of the sensor. Yellow annotations refer to time events and black annotations refer to the main causes of inlier decreases or increases.

2.4 Summary

This chapter presented a high-level overview of the autonomous path following system that this thesis expands upon, Visual Teach & Repeat (VT&R) 1.0 (Furgale and Barfoot, 2010) as well as the state estimation machinery used in this system. We show that the sytem's reliance on tracking visual features from grayscale images between a live stereo image stream and a static taught map severely limits the system in unstructured, outdoor environments. In the following chapters we present novel localization and mapping techniques to address this limitation and provide a teach-and-repeat system capable of long-term operation across multiple seasons. We remind the readers that this chapter provides background information only and presents no novel contributions.

Chapter 3

Multi-Channel Localization

In this chapter, we increase the operational window of the VT&R 1.0 system from a few hours to multiple days in challenging outdoor environments with Multi-Channel Localization (MCL), a novel many-to-one localization and mapping framework. We present two novel localizers based on the MCL framework that are designed to increase autonomy across intra-seasonal appearance change through color-constant imagery and multiple stereo cameras, respectively. Furthermore, we integrate the MCL framework into the VT&R 1.0 system and validate the localizers with extensive field tests covering over 26 km of vision-in-the-loop autonomous driving.

3.1 Introduction

The VT&R 1.0 system enables mobile robots to autonomously traverse large-scale trees of paths previously demonstrated by human operators using only a stereo camera. VT&R 1.0 allows mobile robots to navigate autonomously through large-scale environments using inexpensive commercial sensors without the need for an accurate global map (Furgale and Barfoot, 2010). This technology opens the door for many applications that benefit from repeated traversals over constrained paths such as factory floors, orchards, mines, and urban road networks. Furthermore, this method can be used in hazardous-exploration, sample-return, and search-and-rescue missions where the robot can autonomously return to a previously driven location without the need for a globally consistent map. However, in order for these applications to succeed, robots must have the ability to navigate reliably through their environments over long periods of time. This poses a serious problem for robots that operate in outdoor environments where lighting, weather, and seasonal change dramatically alter the appearance of the scene. An example of daily appearance change can be seen in Figure 3.1, which shows the varying appearance of one of our primary field testing sites due to lighting change. Environments with variable appearance are difficult for autonomous path-following systems, which require vision-in-the-loop navigation. This specific task relies on a vision system to provide continuous, accurate, metric localization to the control loop to keep the robot driving. Most current methods that meet this criterion are examples of single-channel, single-experience localization systems. In the context of this work, we use the term *channel* as a stream of information used to localize a robot's position and *experience* as a collection of channel data obtained during a robot traverse, or run. Methods that rely on localizing to a map collected from a single experience with a single channel of vision information are highly susceptible to appearance change. As a result, the operational domain of these methods are typically limited to only a few hours outdoors, as highlighted by Furgale and Barfoot (2010), due directly to appearance change.

(a) Impact of the sun in forest environments. Tall trees cast shadows that can envelope the entire scene. This sequence shows the same scene at different times of day, namely, 10:58, 13:10, 15:17, and 17:44.

(b) Impact of the sun and robot in desert environments. Small textures such as rocks and wavy sand cast shadows and vehicles driving in sand significantly alter the terrain in a short amount of time. This sequence shows the same scene at different times of day, namely, at 11:03, 13:15, 15:12, 17:49

Figure 3.1: Examples of the daily appearance change seen in unstructured, outdoor environments due to lighting change and terrain modification.

This chapter addresses the specific challenge of intra-seasonal, daily appearance change and provides solutions that extend the operational domain of path-following algorithms from a few hours to multiple days. This is achieved through the use of the novel Multi-Channel Localization (MCL) framework, where multiple information channels are used together to solve a single state estimation problem. Different channels can emerge from the same sensor, from the same type of sensor, or from different visual sensors. An example of a channel in the context of a localizer that uses sparse visual features with depth is the stream of grayscale image data from a stereo camera. We integrate the MCL framework into the VT&R 1.0 path-following system in order to gain robustness against environmental changes and provide two concrete instantiations: a lighting-resistant variant that uses environmentally tuned color-constant images (Paton et al., 2015a), and a variant with an extended field of view through multiple stereo cameras (Paton et al., 2015b). To experimentally validate each localizer's capability to provide autonomous path following across intra-seasonal appearance change, we conducted three field tests in harsh, outdoor conditions, covering over 26 km of vision-in-the-loop autonomy. Two of these field tests were conducted at the peak of Canadian winter, experiencing difficult conditions such as low sun elevation and snow. We conclude the chapter with a detailed analysis on the contrast in vision-based localization performance between summer and winter environments.

To summarize, the novel contributions of this chapter are: i) a multi-channel localization framework that performs independent tracking of point-based visual features for multiple information channels and fuses data correspondences from all channels into a single state estimation problem, ii) a lighting resistant localization system that uses the multi-channel framework to fuse data correspondences from grayscale images and color-constant images to improve performance across lighting change, iii) a multi-stereo localization system that uses the multi-channel framework to fuse data correspondences from multiple stereo cameras to increase the field of view of the localization system and improve performance across general appearance change, iv) an in-depth analysis of expected localization performance in varying

seasons with insight on the limitations of single-experience localization systems that rely on point-based visual features in difficult winter environments, and v) a methodology to experimentally tune the color-constant image transformations to improve performance in a given environment with respect to visual features tracked across lighting change. The contributions in this chapter have appeared in three conference papers and one journal paper. The lighting-resistant localizer was published in the proceedings of the International Conference on Robotics and Automation (ICRA) (Paton et al., 2015a), the multi-stereo localizer was published in the proceedings of the Canadian Conference on Computer and Robot Vision (CRV) (Paton et al., 2015b), field results of both systems, including insight into expected performance in harsh winter conditions was published in the proceedings of the international conference on Field and Service Robotics (FSR) (Paton et al., 2015c). Finally, the MCL framework was formalized with both localization systems as well as detailed analyses of all field tests as a journal article in the special edition on Field and Service Robotics in the Journal of Field Robotics (JFR) (Paton et al., 2017b).

3.2 Related Work

This chapter presents MCL, a multi-channel localization and mapping framework designed to increase the performance of vision-based metric localization between two views across intra-seasonal appearance change due primarily to lighting. This framework is used to increase localization robustness against lighting change through the use of color-constant images and multiple stereo cameras. These localization methods are used to extend the performance of the autonomous path-following system, VT&R 1.0, to increase its operational window from a few hours to multiple days in unstructured, outdoor environments. We furthermore explore the trends related to a decrease in localization performance in winter environments. As such, work related to this chapter spans the following topics: (i) autonomous path-following systems, (ii) color-constancy in robotics, (iii) localization in dynamic environments, (iv) localization and VO using multiple cameras, and finally (v) localization and VO in extreme environments.

3.2.1 Autonomous Path Following Systems

The novel contributions presented in this chapter are extensions to the existing autonomous path-following system, VT&R 1.0, summarized as background information in Section 2.3. The VT&R 1.0 system first published by Furgale and Barfoot (2010), provides short-term, vision-based path following on large-scale tree structures. They demonstrate the system's ability to autonomously navigate large-scale terrain through extensive field tests, including a 3.2 km autonomous traverse in a Mars analogue environment in the Canadian high arctic. The VT&R 1.0 system is furthermore capable of autonomous exploration through the Network of Reusable Paths (NRP) algorithm (Stenning et al., 2013). This algorithm exploits the system's ability to autonomously repeat previously driven paths to build a traversable tree of reusable paths between the starting point and a goal location through terrain assessment, autonomous retrotraverse, and exploration using Rapidly-exploring Random Tree (RRT)s. While effective at long-range navigation, the VT&R 1.0 system is highly susceptible to lighting change while operating outdoors, limiting successful operation to a window of only a few hours. This is primarily due to the decay of point-based visual feature associations between the live view and the map as the appearance of the scene changes. Apart from VT&R 1.0, short-term, vision-based path-following systems have been demonstrated using heading-only navigation by Chen and Birchfield (2009) and Krajnk et al. (2010).

In these monocular-vision systems, data correspondences between the live view and map are used to estimate the heading of the rover with respect to the path being traversed, with wheel odometry used to estimate translation. Despite the fusion of wheel odometry, the lack of a reliable translation estimate makes navigation in constrained environments unsafe.

One method of overcoming the issue of appearance change in autonomous path-following systems is through the use of an active sensor, which is inherently invariant to external lighting conditions, providing long-term navigation. The VT&R 1.0 system was reformulated to use an appearance-based LiDAR sensor that produces intensity images with depth information in place of a stereo camera (McManus et al., 2012). This extension to the VT&R 1.0 system allows for reliable autonomous navigation regardless of the external lighting of the scene, including the total absence of light at night or underground. However, this formulation of the system suffers from motion blur issues without additional state estimation techniques to compensate and relies on an expensive sensor that is commercially unavailable. Krüsi et al. (2014) perform autonomous path following through dense point-cloud registration. Their method demonstrates accurate path following across lighting change using a 3D scanning LiDAR and includes local re-planning to avoid obstacles on the followed path. While active sensors are better suited to long-term navigation than passive sensors, there are compelling reasons for long-term path-following systems that rely only on passive sensors. Passive sensors are commercially available at low costs, require a small amount of power, and contain little to no moving parts. Furthermore, vision sensors provide a texture-rich view of the scene, which can be used for tasks such as semantic segmentation and terrain classification. These advantages come at the cost of being highly susceptible to changes in lighting. One method to overcome this issue is the use of color-constant images.

3.2.2 Color-Constancy Theory

The localizers presented in this system rely on color-constant images to achieve partial invariance to lighting conditions. Color-constancy can be defined as the ability to observe an object's color largely independent of the varying illumination. It is a property our our human perceptual system and has been a topic of research in the optics and computer vision communities. Recent research has developed simple, fast transformations from RGB images to grayscale images that are partially invariant to lighting conditions. If assumptions are made about the sensor and the light source, a grayscale, lighting-invariant image can be obtained from a three-channel camera. Finlayson et al. (2006) calculate a 2D colorspace that moves along a known direction as the lighting in the environment changes. By projecting the 2D colorspace onto a line that is orthogonal to this direction, a 1D colorspace that is invariant to lighting conditions is obtained. Ratnasingam and Collins (2010) extract a grayscale image by taking the weighted log difference of the three channels to cancel out the effects of illumination. More details on this technique can be found in Section 3.3.3.

3.2.3 Localization across Intra-Seasonal Appearance Change

Lighting change is the first issue vision-based navigation systems need to face when operating outdoors. Fortunately, the theories of color constancy have had great success increasing the robustness of vision-based localization and place recognition systems. Corke et al. (2013) tested the image transformation described by Finlayson et al. (2006) on a data set of images captured under varying illumination conditions. They show an increase in precision/recall performance versus grayscale images when whole-image

place recognition is performed on this data set. MacTavish et al. (2015) further showed that color-constant images boost the performance of feature-based place-recognition systems such as FAB-MAP (Cummins and Newman, 2008). Maddern et al. (2014) localize monocular images against a prior map of colored 3D point clouds generated from the fusion of a monocular camera and a LiDAR to obtain a 6DOF pose estimate. By using color-constant images based on Ratnasingam and Collins (2010) during the day and grayscale images at night, they show successful localization over a 24-hour period. McManus et al. (2014a) run two separate localizers in parallel, one that uses grayscale images and one that uses color-constant images based on Ratnasingam and Collins (2010). Localization with color-constant images only occurs when the grayscale localizer first fails. This work was shown to improve localization against a map that was collected in different lighting conditions. This *Best Fit* approach is directly compared in Section 3.6.1 to our multi-channel framework (Section 3.3.3). We show an increase in localization performance using the MCL framework by combining data correspondences from the color-constant image transformations detailed in Section 3.3.3 and grayscale images to solve a single state estimation problem.

3.2.4 Localization and VO using multiple cameras

Work on multi-camera state estimation can be broken into two categories: systems that model multiple cameras as a single generalized camera, and systems that treat each camera independently. Our MEL-based multi-stereo localizer (Section 3.3.4) falls into the latter category, where multiple stereo cameras are treated independently, and are used together to solve for a single pose estimate.

Systems that model multiple cameras as one are typically based on the *generalized camera model* formulated by Pless (2003). The use of plücker lines to model point correspondences between cameras allow this model to solve for extrinsic calibration parameters using a generalized essential matrix. Lee et al. (2013) estimate the motion of self-driving cars with four non-overlapping monocular cameras. With inter- and intra-point correspondences, they solve for the generalized essential matrix with a 2-Point RANSAC scheme and nonlinear refinement. Heng et al. (2014) present a full Micro Aerial Vehicle (MAV) SLAM system including autonomous calibration of extrinsic parameters between cameras. Their system setup consists of an Inertial Measurement Unit (IMU) and four monocular cameras placed in a dual-stereo configuration. To calibrate, they fly the vehicle in a pattern while performing dual-stereo bundle adjustment. The generalized camera model is used for the SLAM problem, when all four cameras are treated as one. Kneip et al. (2013) formulate a general solution to multi-camera state estimation that is computationally more efficient than previous methods. They present a parametrization of the generalized camera model that is non-iterative and linear in complexity with respect to the number of points. They show tests in simulation and on a real camera system.

The alternative to a generalized camera model for multi-camera state estimation systems is to formulate the system as a set of independent camera sensors. Oskiper et al. (2007) perform VO using a dual-stereo rig and an IMU. Motion is estimated through independent stereo pipelines. Using known extrinsic parameters, pose estimates from each camera are evaluated on *all* point correspondences. At each step, the estimate with the smallest reprojection error is used. Clipp et al. (2008) build a 6DOF motion estimation system using clusters of non-overlapping monocular cameras. Each camera performs independent state estimation through a 5-Point RANSAC algorithm. Any inter-camera correspondences are then used to solve for scale using a 1-Point RANSAC solution. At each step, the best estimate is used. Kazik et al. (2012) estimate motion with two non-overlapping monocular cameras. Monocular VO is first performed individually on each camera up to scale. Enforcing the known transforms between

cameras, they derive a linear least-squares problem to solve for the scale of the VO transformations on each camera. They use multi-frame estimation to improve accuracy. Motion estimates from each camera are then fused to obtain the final 6DOF motion estimate.

Our multi-stereo system is similar to the dual-stereo VO setup described in Oskiper et al. (2007), with the exception that we use point correspondences from both stereo cameras to form a single pose estimate. This allows for the minimum number of required keypoints to be spread across both cameras, allowing for localization in keypoint-limited environments. More similar to our method is the Non-Overlapping Multi-Camera Parallel Tracking and Mapping (MCPTAM) algorithm formulated in Tribou et al. (2015). In this method, localization is achieved by running two parallel processing threads: one for frame-to-keyframe VO, and one for full keyframe bundle adjustment. While effective, this method's reliance on a single privileged reference frame and a global bundle adjustment solution is not well suited for large-scale, outdoor navigation targeted in autonomous path-following applications.

3.2.5 Localization and VO in Extreme Environments

The performance of vision-based state-estimation systems is in part, dependent on the environment in which the robot is operating. Two environment-dependent factors significantly affect vision-based systems: the rate of appearance change and the amount of contrast in the scene. This makes vision-based navigation in winter environments especially difficult—as the elevation of the sun is perpetually low on the horizon, and snow rapidly accumulates, melts, and provides little contrast to the scene. Williams and Howard (2010) improve VO in snowy environments by applying Contrast Limited Adaptive Histogram Equalization (CLAHE) to increase keypoint matches in images with snowy foregrounds. They show an increase in keypoint match count by an order of magnitude. Volcanic fields are similar to snowy landscapes in their lack of contrast. Otsu et al. (2015) extract and track different keypoints depending on the volcanic terrain, they show an improvement in keypoint count and computation speed. An often ignored environment for vision sensors is night. Nelson et al. (2015) perform vision-based, night-time localization through the tracking of artificial light sources such as street lights.

3.3 Methodology

This section presents the details of the MCL-based VT&R system, which uses parallel information channels to increase robustness of metric localization across appearance change. The section starts with a sensor-generic formulation of the MCL system. Next, we present two VT&R systems that use the MCL framework: (i) the lighting-resistant VT&R system originally published in Paton et al. (2015a), and (ii) the dual-stereo VT&R system originally published in Paton et al. (2015b).

3.3.1 System Overview

Map Representation The MCL system map shown in Figure 3.2 is a modified version of the VT&R 1.0 topometric pose graph (Section 2.3). Vertices (black triangles) in this graph structure represent a robot's pose at a key time and store raw sensor observations and triangulated 3D landmarks with associated descriptors observed at that time from *each* information channel. A channel in this context is a stream of sensor information capable of producing these landmarks. In the example figure, there

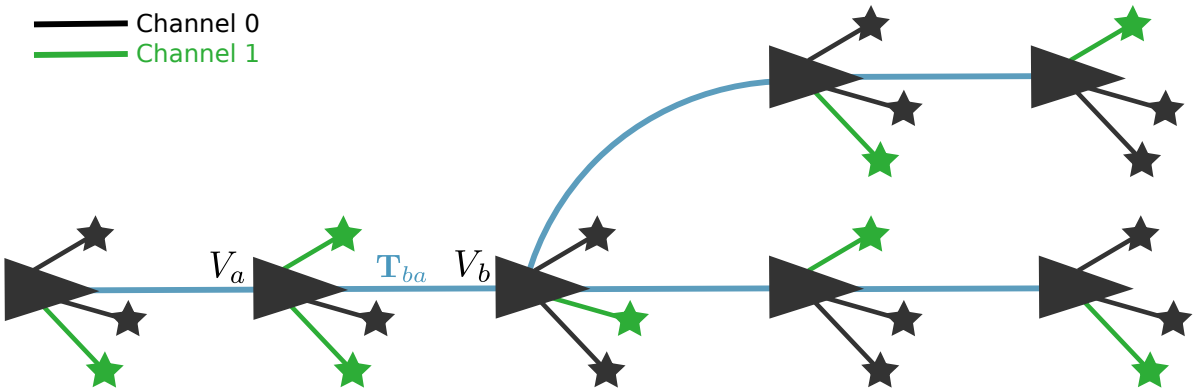


Figure 3.2: Overview of the topometric pose graph used to represent the tree of paths used by the MCL system. Vertices (black triangles) in the graph represent a robot's pose at a key time and contain triangulated stereo landmarks from each information channel (black and green stars) with 3D positions and descriptors. Vertices are connected through edges (blue lines) and are related metrically with a relative, $SE(3)$ transformation. This map structure is built during demonstration of the path by a human operator in the teaching phase. Once established, the robot can use this data structure to autonomously follow a route through the graph.

are landmarks from two information channels, represented by green and black stars, respectively. Channels are assumed to be generated from sensors that are temporally synchronized with known extrinsic transformations between each other. Landmark positions in the graph are represented in the coordinate frame of their respective channel sensor at the time the vertex was created. Edges in the graph (blue, lines) link vertices metrically with a relative $SE(3)$ transformation.

Teaching Phase During the teach phase, the robot is manually driven while building the map represented by the multi-channel, topometric pose graph detailed in the previous paragraph. Vertices in the map are constructed when the robot's motion exceeds a specified threshold, forcing an evenly distributed map. To build a branch from the main path in the map, the user can command the robot to autonomously traverse the path to a desired branching point while in the repeating phase, and then command the robot to begin branching. At this point, a localization problem using the multi-channel state estimation pipeline is performed to compute the robot's position with respect to the closest vertex in the map. This creates the first vertex of the branch as well as the relative, $SE(3)$ transformation between the two vertices. After this process is complete the user may drive the robot to begin creating the new branch.

Repeating Phase To autonomously repeat the taught path, the robot performs VO and localization to obtain a relative transformation between the current position and the map. Localization is achieved by comparing the stream of multi-channel data to a local submap pulled from the closest vertex. This submap is computed from a fixed number of vertices centered at the estimated closest vertex and relaxed into a single coordinate frame. Doing this allows the localization complexity to be constant with respect to the size of the total map. In the case of a localization success, the VO solution is used as a prior, in the case of a localization failure, the VO solution is propagated from the last localization estimate. This information is fed to a path-tracking controller to keep the robot on the path. Path tracking is accomplished using Model Predictive Control (Rawlings and Mayne, 2009). At the start of a repeat, the robot performs a localization search to find its position relative to the closest vertex in the pose graph.

3.3.2 Multi-Channel Localization (MCL)

The multi-channel state estimation system, depicted in Figure 3.3, increases robustness against appearance change by combining landmarks from multiple channels of visual information into a single state estimation problem. This process can be broken into two steps: Independent Channel Tracking and Multi-Channel State Estimation.

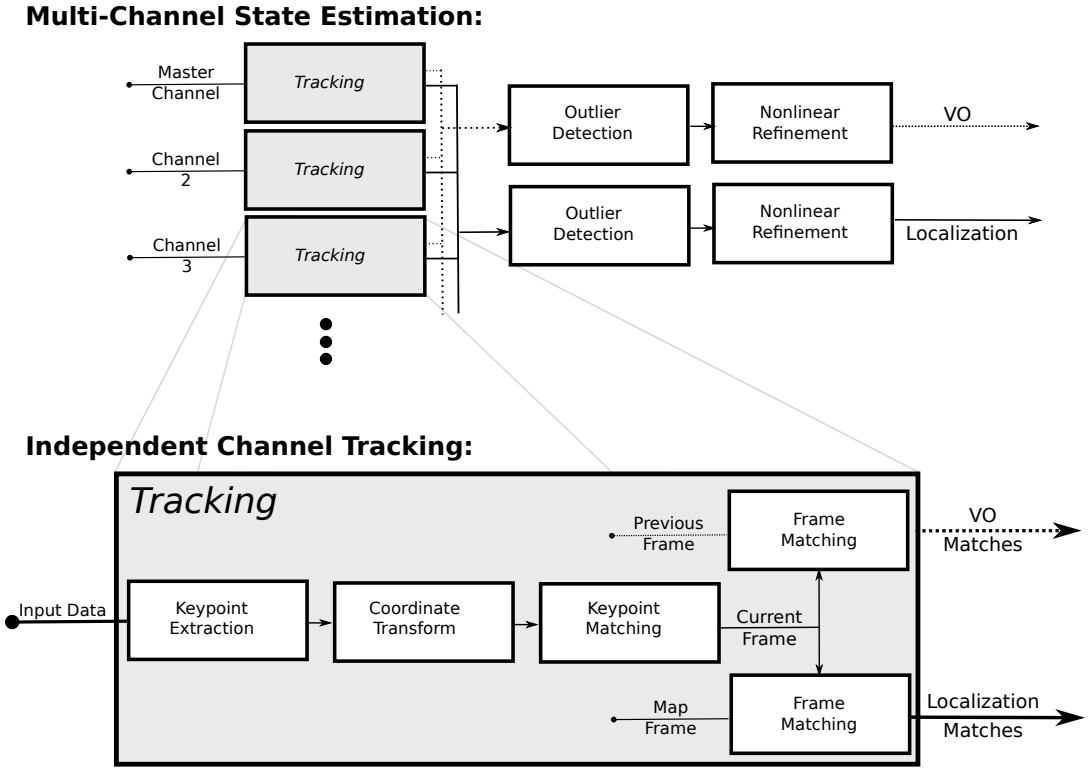


Figure 3.3: Pipelines of the multi-channel state-estimation system. *Top:* Multi-Channel State Estimation Pipeline. The input to the system is a set of synchronized data from all channels. The output to the system is a transformation relative to a reference frame. Each channel performs independent tracking to obtain data correspondences in the coordinate frame of the master channel. These correspondences are then fused together to solve for the relative motion of the master channel through outlier rejection and nonlinear refinement. *Bottom:* Independent channel tracking. The input to the system is vision data with the ability to extract depth information. The output to the system is a set of keypoint matches with depth in the coordinate frame of the master channel. Keypoints are matched between either the previous frame in the case of VO, or the map in the case of localization to obtain a set of data correspondences.

Independent Channel Tracking In the context of this system, a *channel* defines a stream of visual information used to localize a robot's position. A key innovation behind the multi-channel VT&R paradigm is that landmarks independently detected and tracked in channels can be used to solve a single state estimation problem. The MCL system can use channels from the same sensor (i.e., grayscale images, color-constant images), from the same type of sensor (multiple cameras), or different sensors (stereo and lidar). In this thesis, we focus on channels that originate from stereo cameras. Multi-channel state estimation requires the channels to be temporally synchronized with sensor transforms known *a priori*. Channel tracking consists of the detection and matching of point-based descriptors. This process

is detailed in the lower section of Figure 3.3. The input to the system is visual data with the ability to extract depth information. The output is a set of data correspondences between the input and either the previous frame (VO) or a map frame (localization).

The first step of channel tracking is the extraction of keypoints with descriptors, 3D position, and uncertainty associated with the image measurement. This algorithm is agnostic to the methods associated with depth extraction, keypoint detection, and keypoint description. Coordinates of the j^{th} keypoint at time k are of the following form:

$$\mathbf{y}_{j,k_i} = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix}, \qquad \mathbf{p}_{k_i}^{j,k_i} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.1}$$

where $\mathbf{y}_{j,k_i}$ represents the left-right keypoint measurement coordinates and $\mathbf{p}_{k_i}^{j,k_i}$ is the physical location of landmark j, in homogeneous coordinates, in the coordinate frame of channel i. This value is a vector from the origin of $\underrightarrow{\mathcal{F}}_{k_i}$ to the origin of $\underrightarrow{\mathcal{F}}_{j}$ (denoted by the superscript) and expressed in $\underrightarrow{\mathcal{F}}_{k_i}$ (denoted by the subscript).

In order to fuse data correspondences between channels, they must be in the same coordinate frame. Because all state estimation is performed in the reference frame of the single master channel, keypoints in the coordinate frame of the ith channel, $\underrightarrow{\mathcal{F}}_{k_i}$, are converted to the coordinate frame of the master channel, $\underrightarrow{\mathcal{F}}_{k_1}$:

$$\mathbf{p}_{k_1}^{j,k_i} = \mathbf{T}_{1,i}\mathbf{p}_{k_i}^{j,k_i}, \tag{3.2}$$

where $\mathbf{T}_{1,i}$ is the extrinsic transformation between channel i and channel 1, assumed to be known *a priori*. Keypoints in the coordinate frame of the master channel remain unmodified. The final step of the channel tracking process is to match keypoints from the current view to either the previous frame in the case of VO, or the map in the case of localization. In both cases, the end result is a list of corresponding keypoints with 3D position information in the coordinate frame of the master channel.

Multi-Channel Estimation Framework The goal of both localization and VO is to estimate the relative motion of the master-channel sensor between the current view at time k, $\underrightarrow{\mathcal{F}}_{k_1}$, and a reference frame, $\underrightarrow{\mathcal{F}}_{m_1}$. This motion can be represented by a transformation matrix, $\mathbf{T}_{k_1,m_1}$, which takes points from $\underrightarrow{\mathcal{F}}_{m_1}$ into $\underrightarrow{\mathcal{F}}_{k_1}$. In the case of VO, the reference frame is the previous frame while in the case of localization, the reference frame is a local submap. In both cases, we wish to find the estimate of $\mathbf{T}_{k_1,m_1}$ that minimizes the reprojection error of all of the landmark observations after they are transformed and reprojected into the image plane. For a given keypoint measurement of landmark j, $\mathbf{y}_{j,k}$, and an observation of the landmark from the reference frame, $\mathbf{p}_{m_1}^{j,m_1}$, the error term, $\mathbf{e}_{j,k}$ is given by

$$\mathbf{e}_{j,k} = \mathbf{y}_{j,k} - \mathbf{g}(\mathbf{T}_{k_1,m_1}\mathbf{p}_{m_1}^{j,m_1}), \tag{3.3}$$

where $\mathbf{g}(\cdot)$, is the stereo observation model that reprojects points into the image sensor plane. Each keypoint also contains an uncertainty, $\mathbf{Q}_j$, of the measurement of landmark j.

The localization-and-VO pipeline is depicted in the upper half of Figure 3.3 and consists of the fol-

lowing steps: (i) Channel Tracking, (ii) Outlier Rejection, and (iii) Nonlinear Refinement. The inputs to the state estimation system are sets of image data from each channel. Each channel first undergoes keypoint tracking to obtain data correspondences. Because correspondences from all channels are formulated in the reference frame of the master channel they can be concatenated and sent to an outlier rejection algorithm.

Keypoints tracked by all channels are sent through a RANSAC implementation using Horn's 3-point method (Horn, 1987). This provides a set of inliers as well as an initial estimate of the master channel's pose. The goal of the solver is to minimize the following objective function with respect to the camera transformation, $\mathbf{T}_{k_1,m_1}$:

$$J_k = \frac{1}{2}\sum_{j=1}^{n} \mathbf{e}_{j,k}^T \mathbf{Q}_j^{-1} \mathbf{e}_{j,k} + J_{pos}, \tag{3.4}$$

where $(\mathbf{e}_{1,k}, \ldots, \mathbf{e}_{n,k})$ is the set of errors associated with data correspondences from all channels, and J_{pos} is a prior term on motion. J_{pos} minimizes the error between the posterior transform, $\mathbf{T}_{k_1,m_1}$, and a prior transform, $\check{\mathbf{T}}_{k_1,m_1}$. In the case of VO, $\check{\mathbf{T}}_{k_1,m_1}$ is a no-motion prior, and in the case of localization, $\check{\mathbf{T}}_{k_1,m_1}$ is the result of VO. To minimize this objective function, the equation is linearized and then iteratively refined through the Levenberg-Marqardt algorithm. The result is a transformation that minimizes the sum of reprojection errors in all channels.

3.3.3 Lighting-Resistant Localization

This section presents the lighting-resistant localization algorithm originally published in Paton et al. (2015a). This algorithm makes use of the MCL framework to fuse data correspondences from traditional grayscale stereo images and color-constant stereo images, which are partially invariant to outdoor lighting conditions.

Color-Constancy Theory

We introduce the theory of the color-constant image transformations used in this thesis at a high level, and refer the reader to Ratnasingam and Collins (2010) for a detailed derivation. A camera's response for a specific point, x, in the environment is described by the illuminant, sensor response, reflecting surface, and the geometry of the scene and camera. The light originates from an illuminant, is reflected by a surface towards the camera, and is focused onto an image sensor consisting of an array of filtered pixel sensors. This process results in the sensor response, R^x, describing the power of the light incident on the pixel sensor after being reflected and filtered. The illuminant is described by its intensity, I, and spectral power distribution, $E(\lambda, T)$, as a function of wavelength, λ, and temperature, T. At a specific point, x, the light is reflected according to the incident direction, $\underrightarrow{a}^x$, the surface normal, $\underrightarrow{n}^x$, and the surface reflectance, $S^x(\lambda)$. This light is filtered according to the sensor's channel, described by the spectral sensitivity, $F(\lambda)$. Integrating over the desired spectrum, ω, results in the image sensor response:

$$R^x = \underrightarrow{a}^x \cdot \underrightarrow{n}^x I \int_{\omega} S^x(\lambda) E(\lambda, T) F(\lambda) d\lambda. \tag{3.5}$$

Images that are resistant to the variation in the illumination of an outdoor scene can be calculated from a three-channel camera by making assumptions about the imaging sensor and environment (Ratnasingam

and Collins, 2010). These assumptions allow cancellation of the factors of (3.5) that are dependent on the scene's illumination: the spectral power distribution, $E(\lambda, T)$, and the intensity of the illuminant, I. If the assumptions that the spectral sensitivity function, $F(\lambda)$, is infinitely narrow at the sensor's peak wavelength, λ_i, and the sole illuminant of the scene is a black-body radiator, then the logarithm of (3.5) can be reformulated as:

$$\log(R_i^x) = \log(\underrightarrow{a}^x \cdot \underrightarrow{n}^x I) + \log(S^x(\lambda_i) C_1 \lambda_i^{-5}) - \frac{C_2}{T\lambda_i}, \tag{3.6}$$

where C_1 and C_2 are constants. The result is a sensor response equation that separates the effect of illumination on the scene from properties of the reflected surface material. A weighted linear combination of three channel responses can be constructed to effectively cancel out the first and third terms, providing an illumination-invariant sensor response that is affected only by the properties of the surface materials. This difference of log responses is provided on a per-pixel basis by the following equation:

$$F = \log(R_2) - \alpha \log(R_1) - \beta \log(R_3), \tag{3.7}$$

where $\log(R_i)$ is (3.6) with peak wavelength, λ_i, and weights α and β subject to the following constraints:

$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{\beta}{\lambda_3}, \quad \beta = (1 - \alpha). \tag{3.8}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the theoretical peak sensor responses ordered from lowest to highest wavelength. If these constraints are met, the weighted difference of the log responses will cancel out the effect of the spectral power distribution of the light source, $E(\lambda)$, and the illuminant intensity, $\underrightarrow{a}^x \cdot \underrightarrow{n}^x I$. In the context of a digital, RGB camera, $\{R_1, R_2, R_3\}$ are the pixel response values for the blue green and red channels, respectively.

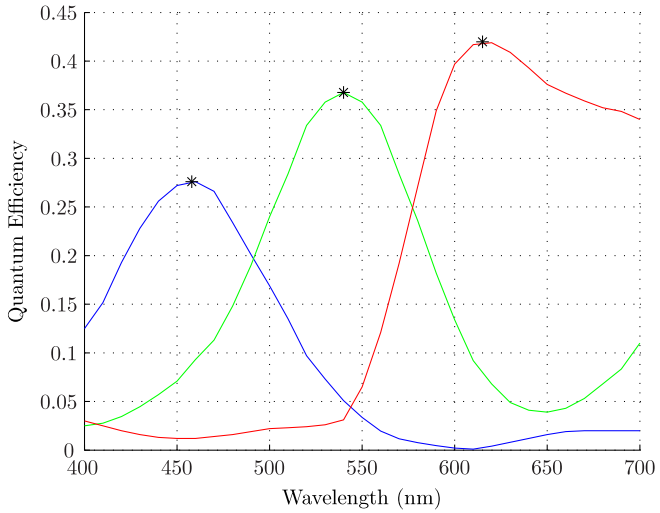


Figure 3.4: Sensor response of the Sony ICX445 CCD, with theoretical peak wavelengths denoted by stars.

In theory, if the peak wavelength values of the sensor, $(\lambda_1, \lambda_2, \lambda_3)$, are known, then the weights can

be calculated based on the constraints in (3.8) with the following:

$$\alpha = \frac{(\lambda_1\lambda_3)/\lambda_2 - \lambda_1}{\lambda_3 - \lambda_1}, \quad \beta = 1 - \alpha \tag{3.9}$$

If the assumptions are met that the sole illuminant of the scene is a black-body radiator and the sensor channels of the camera are infinitely narrow, centered at their peak values, then the resulting image will be free of the effects of illumination. The first assumption is reasonably close to the truth, as long as the only illuminant is the sun. The accuracy of the second assumption varies for each camera, but will never be exactly true. An example of this can be seen in the sensor response curves for the Sony ICX446 CCD imaging sensor (see Figure 3.4). While each channel has distinct peaks, they are far from infinitely narrow with significant overlap between channels. Using the theoretical wavelengths seen in the response curves, the color-constant image transformation would be Equation 3.7 with weights $\alpha = 0.467$ and $\beta = 0.533$. While these parameters have a theoretical basis, they do not always produce the best results. In Section 3.3.3 a method to find the color-constant transformation parameters that provide the best results for specific biomes is detailed.

Environment-Dependent Color Constancy

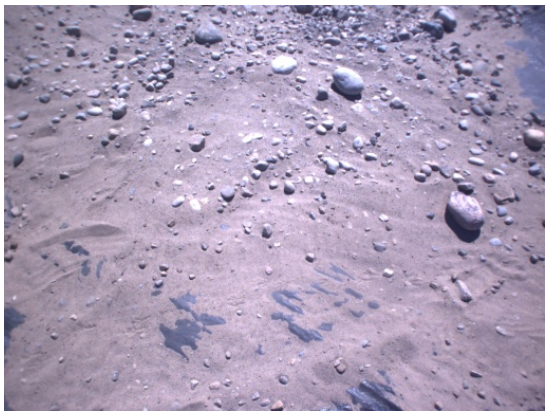
In theory, the color-constant image transformation for a given camera can be computed by plugging the peak values from the camera's spectral response curve into Equation 3.9. However, the assumption that senor's channel responses are infinitely narrow at their peaks is violated to varying degrees for typical, 3-channel RGB cameras. This can be seen in Figure 3.4, which shows the response curves for the Sony ICX445 CCD, the sensor used throughout this thesis. It can be seen that for this sensor, the response curves are wide and overlapping with each other. This section presents a simple empirical method to tuning the weights of the color-constant image transformation to increase point-based visual feature matching in a *given environment*. In the context of localization, these color-constant image transformations are used to increase the number of descriptor matches across lighting change. Therefore, we are motivated to find the set of weights that yields the best performance with regards to keypoint detection and matching.

This can be achieved through a series of static time-lapse experiments. With a collection of stereo time-lapse data across significant lighting change, we can search for the set of weights that maximizes the number of keypoint correspondences between images. This search can be achieved by relaxing the first constraint in (3.8). We decided to relax the first constraint for two reasons: (i) this constraint is based on the assumption of infinitely narrow peak wavelength responses, which is far from the truth for typical sensors and (ii) the peak wavelengths needed to calculate the weights are often unknown. Relaxing the constraint provides a free variable, α, with the weight, $\beta = 1 - \alpha$. From here, we can perform a brute force search of a discrete set of α values centered at zero to find a value that maximizes visual feature matching across lighting change.

Static Timelapse Imagery We performed a static experiment in order to tune the color-constant images according to the method proposed in Section 3.3.3, collecting stereo time-lapse imagery from sunrise to sunset in environments related to *Forest* and *Desert* biomes. Figure 3.5 shows key examples of those two biomes. For each data set, a rectified stereo image pair was collected every 10 minutes from sunrise to sunset, resulting in a collection of approximately 60-70 stereo image pairs. The Desert

data set was collected on May 24, 2014, on a sunny day, between the hours of 07:00 and 21:00. The second data set representing a Forest biome was recorded on November 20, 2015, on a partly sunny day, between the hours of 07:00 and 17:00. Over the course of the day, large, fast-moving clouds were passing over the sun causing the scene to constantly switch between sunny and overcast conditions.

(a) Example Image from the Forest static data set.



(b) Example Image from the Desert static data set.

(c) Impact of the sun on the Desert data set. The sequence of images represent the same object at different times of the day, namely 8:00, 10:00, 12:00, 14:00 and 17:00.

(d) Impact of the sun on the Forest data set. The sequence of images represent the same object at different times of the day, namely 8:00, 10:00, 12:00, 14:00 and 16:00.

Figure 3.5: Example images from the static experiments. These experiments were performed to find color-constant image transformations that maximize descriptor matching performance across lighting change in different biomes. For these experiments, two scenes were selected: The first is an environment with green vegetation, which can be associated to a Forest biome, and the second is an environment with rocks and sand, which can be associated to a Desert biome.

To experimentally find the color-constant image transformation that yield the highest number of keypoint matches, we applied the following procedure for each biome. Color-constant images for the time-lapse data were calculated for each α value ranging from -2.0 to 2.0 in increments of 0.1. Each possible image pairing underwent the same localization process detailed in Section 3.3.2, including left-right matching, triangulation, query-map matching, and outlier detection. Inlier matches were then summed to produce a total number of matches for each α value.

Static Timelapse Results Results from the experiment for both biomes are detailed in Figure 3.6, with the theoretical weight value highlighted. It is interesting to note that the Forest and Desert biomes yielded nearly inverse results. The results of the search found two complementary color-constant images,

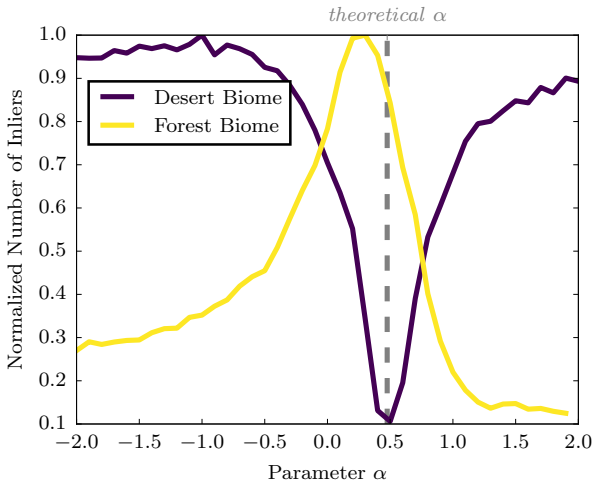


Figure 3.6: Performance of color-constant image transformations for the forest and desert biomes. For each α value, the color-constant image transformation was tested on its ability to perform descriptor matching across lighting change. Note the position of the theoretical peak α value.

Forest CC and *Desert CC*, where Forest CC is (3.7) with weights, $\alpha = 0.3, \beta = 0.7$, and Desert CC is (3.7) with weights, $\alpha = -1.3$, and $\beta = 2.3$. It is worth noting that, while close to the forest biome peak, the theoretical α value of 0.467 produces inferior results in both biomes. A possible explanation is the significant overlap of the sensor response channels seen in Figure 3.4.

These findings are backed by experimental results in Figure 3.7, which shows the gain of inlier matches over the Legacy system (i.e., grayscale images) for both the Forest CC and Desert CC images over all possible image matches in both data sets. It can be seen from the results that the color-constant images that have been tuned for their respective environments outperform both the standard grayscale images and the other color-constant images. A more detailed look at color-constant performance is highlighted in Figure 3.8. This figure shows the number of inlier matches between a reference image and all other images in the experiment for the color-constant images and the Legacy image. Both of the graphs represent the fourth row of the matrices represented in Figure 3.7. It can be seen from the results that the color-constant images that have been tuned for their respective environments outperform both the standard grayscale images and the other color-constant images. While the parameters for the Forest CC image are close to the theoretical transformation, the Desert CC parameters are not. Further investigation into why the Desert CC image performs better than all other images is warranted.

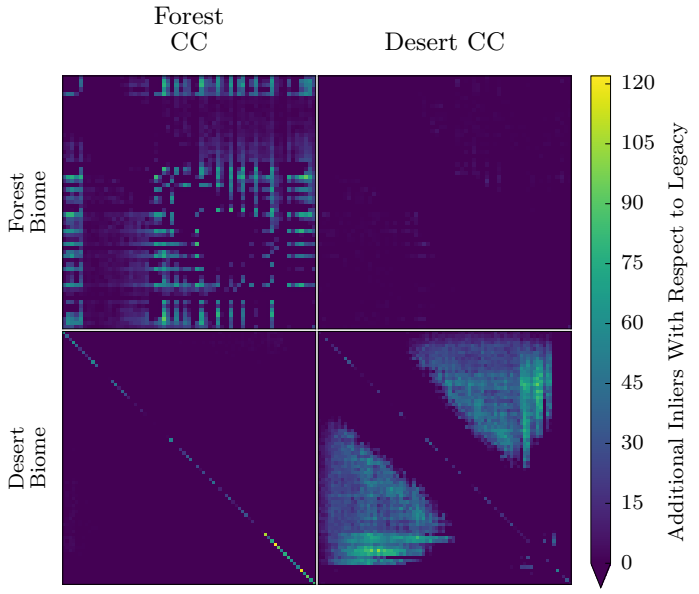
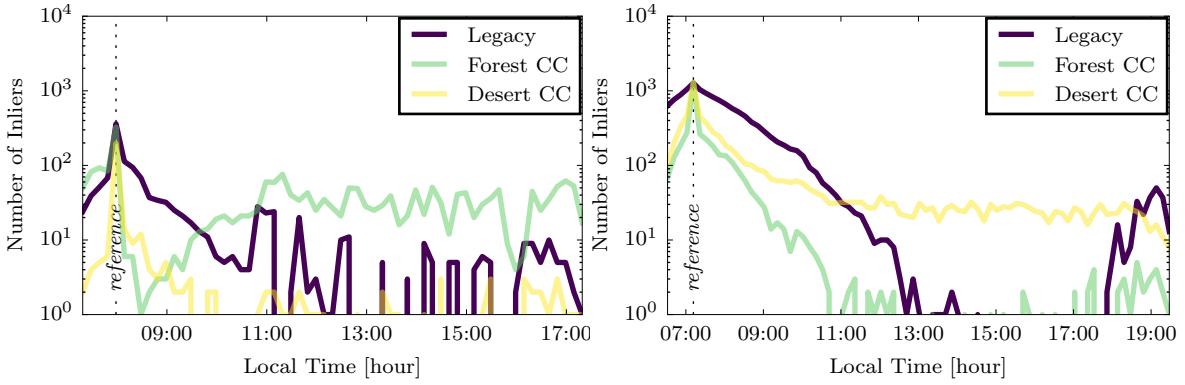


Figure 3.7: Performance gain over the Legacy system for the experimentally tuned color-constant images for each biome. Each quadrant represents the results of the specified color-constant image (horizontal label) in the specified biome (vertical label). Rows and columns in each quadrant represent images separated by 10 minutes. The matrix represents the comparison of a given image (rows) with all other images in the data set (columns), subtracted by the number of inlier matches found in the Legacy, grayscale images.



(a) Results for the Forest data set.

(b) Results for the Desert data set.

Figure 3.8: Influence of the predominant biome on the generation of color-constant images. The graphs represent the evolution of matched keypoints though time, with a static image taken every 10 minutes. Results show that the number of inliers for different systems (Legacy, Forest Color Constant and Desert Color Constant) is influenced by the type of biome (Forest and Desert). Note log scale on the y-axis.

System Overview

The lighting-resistant VT&R system, first published in Paton et al. (2015a), adds robustness against lighting change through the use of color-constant images. The algorithm is formulated in a multi-channel paradigm by setting the input to the master channel to grayscale stereo images and the inputs to subsequent channels to color-constant stereo images experimentally tuned for varying biomes. Inputs to all channels originate from the same RGB stereo pair. We posit that the majority of imaging sensors severely violate the assumptions put forward in (3.8) and may produce inferior results if the peak theoretical wavelengths are used. Furthermore, If a robot is traveling across multiple biomes, it may be necessary to provide the algorithm with multiple color-constant transformations to provide reliable localization across lighting changes.

Each channel performs tracking by extracting SURF keypoints with descriptors and 3D positions. Because all of the channels in this system originate from the same sensor, the transformations that take keypoints into the coordinate frame of the master channel can be assumed to be identity. In this system, VO is performed with the master channel only, while localization uses all available channels. This is because color-constant images are inherently noisier than their grayscale counterparts and lighting change is not a factor for VO. Adding additional channels to the state estimation problem of VO will only add computation cost with no benefit.

3.3.4 Multi-Stereo Localization

This section provides details on the multi-stereo localization algorithm, originally published in Paton et al. (2015b), that uses the MCL framework to fuse data correspondences from multiple stereo cameras.

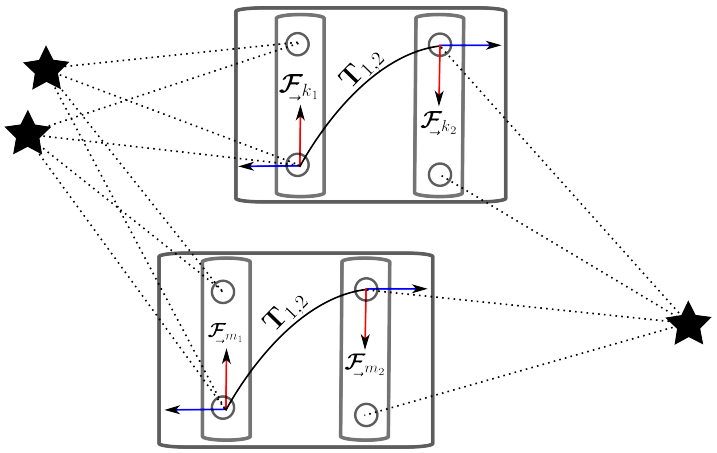


Figure 3.9: Diagram of the multi-stereo setup. Given a timestep, k, the system is defined by a robot with two stereo cameras with respective coordinate frames, $\{\underrightarrow{\mathcal{F}}_{k_1}, \underrightarrow{\mathcal{F}}_{k_2}\}$. The transformation, $\mathbf{T}_{1,2}$, which takes points from $\underrightarrow{\mathcal{F}}_{k_2}$ to $\underrightarrow{\mathcal{F}}_{k_1}$, is assumed be known *a priori*. Localization is achieved through inter-camera point correspondences, depicted as black stars. To localize, all point correspondences are transformed into the frame, $\underrightarrow{\mathcal{F}}_{k_1}$, and used in a joint state estimation problem. Shown here is localization between two timestamps: k and m.

System Overview

The multi-stereo VT&R method, first published in Paton et al. (2015b), extends the field of view of the VT&R 1.0 system through the use of multiple stereo cameras. Using the MCL framework, this localizer provides state estimation through two channels of grayscale image pairs originating from separate stereo cameras. In this formulation, the system assumes two temporally synchronized, non-overlapping, stereo cameras with coordinate frames, $\{\underrightarrow{\mathcal{F}}_{k_1}, \underrightarrow{\mathcal{F}}_{k_2}\}$. The transformation, $\mathbf{T}_{1,2}$, which takes points from $\underrightarrow{\mathcal{F}}_{k_2}$

to $\underrightarrow{\mathcal{F}}_{k_1}$, is assumed be known *a priori*. This algorithm is formulated in a multi-channel paradigm by setting the input to the master channel to grayscale stereo images from camera 1 and setting the input to the additional channel to grayscale stereo images from camera 2. During independent channel tracking, keypoints originating from the second camera are transformed into the coordinate frame of the first camera with $\mathbf{T}_{1,2}$. The camera setup for this system is illustrated in Figure 3.9. While this method is no more resistant to lighting than the original Legacy VT&R method, it essentially doubles the number of inlier matches found during localization, and tracks stable keypoints seen in the environment for longer periods of time. This increase of the algorithm's field of view greatly improves the ability to safely localize in challenging outdoor environments where the appearance rapidly changes. Furthermore, this system is robust to conditions where one of the stereo camera's view is obscured. A common example observed in the field is glare in the image due to the low elevation of the sun.

3.4 Field Tests

This section details the autonomous path-following field tests conducted to experimentally validate the MCL-based localizers. In total, three separate field tests covered over 26 km of driving spanning over a year, covering multiple biomes and seasonal conditions. Overall, our analysis is relying on approximately 1.5 TB of sensor data and 25,000 images.

3.4.1 Hardware

The hardware configuration for the field test is displayed in Figure 3.10. A Clearpath Robotics Grizzly RUV serves as our mobile robot platform. The Grizzly is equipped with a payload that includes a suite of interoceptive and exteroceptive sensors. For the autonomous path-following field tests, the only sensors used for localization and mapping were the forward and rear Point Grey Research (PGR) Bumblebee XB3 stereo cameras labeled in Figure 3.10. The extrinsic transformation between two stereo cameras used in the multi-stereo localization pipeline was hand-measured for these field tests. We collected GPS data during the path for the purpose of visualization only. All of our VT&R code ran on a Lenovo W540 laptop with a Intel® Core™ i7-4800MQ CPU. The static experiment reused the same stereo camera (i.e., PGR Bumblebee XB3 stereo) used on the robot, only mounted on a tripod.

3.4.2 Environments

We conducted a series of extensive field tests to properly test the different variations of the multi-channel VT&R algorithms in realistic, outdoor settings. Over the course of a year, three distinct field tests were performed spanning multiple biomes and seasonal conditions: i) Summer, ii) Winter (no snow), and iii) Winter (with snow). These biomes are on display in Figure 3.11

Summer The first field test was held at the Canadian Space Agency (CSA)'s Mars Emulation Terrain (MET) in Montreal, Quebec, with the purpose of stress testing the Lighting-Resistant VT&R method. The MET is a 60 m by 120 m environment consisting of sand and rocks, emulating the surface of Mars. We chose the MET as an ideal testing environment for our algorithm due to the proximity of rock/sand and grass/forest regions, providing the possibility for a single path to contain both biomes. An example of the forest biome can be seen in Figure 3.11a. To test our algorithm, we taught an approximately 1 km

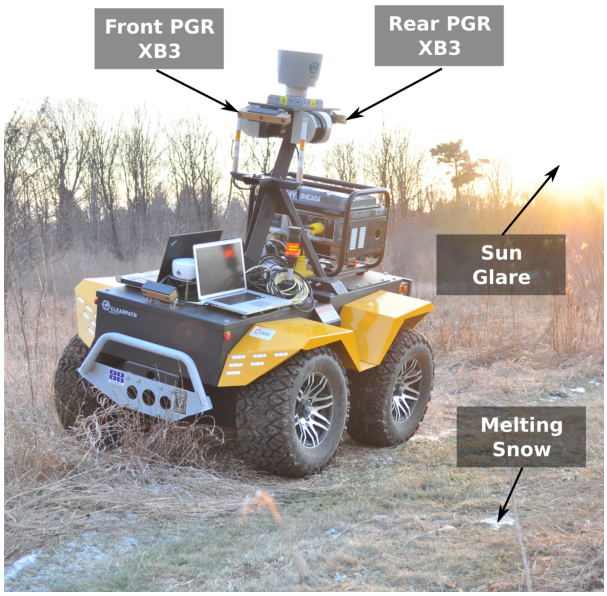


Figure 3.10: Clearpath Grizzly RUV and its sensor configuration. The robot is equipped with a forward- and rear-facing PGR Bumblebee XB3 camera, a GPS receiver, and a Hyundai generator. The robot contains a ROS enabled embedded computer that controls its motors and safety monitors.

path in sunny conditions and repeated the path 26 times over the course of two days, testing localization from sunrise to sunset. Information specific to each repeat is detailed in Table 4.2. An illustration of the sun's elevation, which affects the length of shadows on the scene, for each repeat of the field test can be seen in Figure 3.12.

The 1 km path is displayed in Figure 3.13. It begins in the MET and travels approximately 300 m through rocks and sand before entering into the adjacent field. The path then snakes through the field passing by grass, trees, and a gravel roadway. The path then passes through a wooded area, traveling alongside a stream, re-enters the MET, makes a short loop and finishes at the start of the path. This path was taught at 10:50 am on the first day of the test during bright, sunny conditions with strong shadows. The next two experiments demonstrate the impact of seasonal changes on visual navigation systems as well as investigating the multi-camera VT&R system. Both trajectories are represented in Figure 3.13. We conducted a set of tests in a meadow and a field covered by snow surrounding the UTIAS campus with the purpose of testing the limits of vision-based navigation algorithms in challenging winter environments.

Winter (no snow) This field test was designed to test the multi-stereo system's robustness against lighting change and sun glare in a challenging winter environment. The test occurred in the early winter, before large snow storms covered the entire landscape. Displayed in Figure 3.11b, this environment consists of a large field containing dead vegetation and sparse snow patches surrounded by trees and buildings in the background. Winter environments are difficult for vision systems for a number of reasons: (i) dead vegetation is uniform in color and often matted to the ground, producing little contrast, (ii) tall grass moves with the wind, resulting in keypoint matches that are inconsistent to the movement of the robot, (iii) small patches of snow shrink and change shape as they melt, and (iv) low sun elevation accelerates lighting change between traverses and is often directly in the camera's field of view, which

(a) Summer. (Canadian Space Agency (CSA), Montreal, Canada.)

(b) Winter (University of Toronto, no snow).

(c) Winter (University of Toronto, with snow).

Figure 3.11: Overview of the biomes covered in the autonomous path-following data sets. (a) The CSA MET in the summer which consists of lush vegetation as well as rocks and sand. (a) A Winter meadow consisting of dead vegetation, sparse snow patches, and trees at the horizon. (b) An open field with dead vegetation breaking through a 30 cm snow cover.

Table 3.1: Overview of all autonomous traverses in the MCL field tests.

	ID	Start Time	Duration [hh:mm]	Δt [hh:mm]	Sky Condition	Autonomy [%]	Distance [m]	Nb. Images
Summer	c0	2014/05/12 10:35	00:34	00:00	sunny	Teach Pass	954	32,347
	c1	2014/05/12 11:40	00:28	01:05	sunny	100%	960	25,721
	c2	2014/05/12 12:53	00:27	02:18	sunny	100%	952	24,863
	c3	2014/05/12 13:35	00:26	03:00	sunny	100%	947	24,553
	c4	2014/05/12 14:55	00:31	04:20	sunny	100%	960	30,036
	c5	2014/05/12 16:06	00:32	05:31	cloudy	100%	959	30,323
	c6	2014/05/12 17:27	00:26	06:52	sunny	100%	955	25,520
	c7	2014/05/12 18:14	00:27	07:39	sunny	99.2%	962	25,815
	c9	2014/05/12 19:29	00:25	08:54	sunny	100%	952	23,808
	c10	2014/05/12 20:06	00:25	09:31	sunset	100%	956	24,215
	c11	2014/05/13 06:20	00:28	19:45	cloudy	100%	955	25,126
	c12	2014/05/13 07:05	00:27	20:30	cloudy	100%	956	23,166
	c13	2014/05/13 08:00	00:28	21:25	cloudy	100%	959	23,350
	c14	2014/05/13 09:00	00:25	22:25	cloudy	100%	961	23,830
	c15	2014/05/13 10:00	00:23	23:25	cloudy	100%	955	23,185
	c16	2014/05/13 11:00	00:25	24:25	cloudy	100%	954	23,212
	c17	2014/05/13 12:00	00:29	25:25	cloudy	100%	956	26,847
	c18	2014/05/13 13:00	00:25	26:25	cloudy	100%	944	23,164
	c19	2014/05/13 14:00	00:29	27:25	cloudy	100%	948	24,050
	c20	2014/05/13 15:10	00:26	28:35	cloudy	100%	956	24,587
	c21	2014/05/13 16:00	00:27	29:25	cloudy	100%	951	23,775
	c22	2014/05/13 17:00	00:24	30:25	cloudy	100%	960	23,022
	c23	2014/05/13 18:00	00:32	31:25	cloudy	100%	960	25,582
	c24	2014/05/13 19:00	00:25	32:25	cloudy	100%	959	25,036
	c25	2014/05/13 20:00	00:25	33:25	sunset	99.9%	959	23,686
	c26	2014/05/13 20:30	00:07	33:55	dark	failed	312	9,441
Winter (no snow)	m0	2015/01/28 12:28	00:03	00:00	sunny	Teach Pass	113	7,028
	m1	2015/01/28 12:37	00:03	00:09	sunny	100%	114	6,638
	m2	2015/01/28 15:23	00:04	02:55	sunny	100%	114	7,084
	m3	2015/01/28 15:39	00:04	03:11	sunny	100%	114	7,152
	m4	2015/01/28 16:07	00:03	03:39	sunny	100%	114	5,842
	m5	2015/01/28 16:22	00:04	03:54	sunny	100%	114	6,834
	m6	2015/01/28 16:34	00:03	04:06	sunny	100%	114	5,910
	m7	2015/01/28 16:45	00:04	04:17	sunny	100%	114	6,814
Winter	w0	2015/01/30 13:44	00:06	00:00	sunny	Teach Pass	225	12,868
	w1	2015/01/30 13:55	00:05	00:11	sunny	100%	226	10,204
	w2	2015/01/30 15:57	00:06	02:13	sunny	100%	226	10,446
Total:	38	—	12h 11m	—	—	99.96%	26k	725k

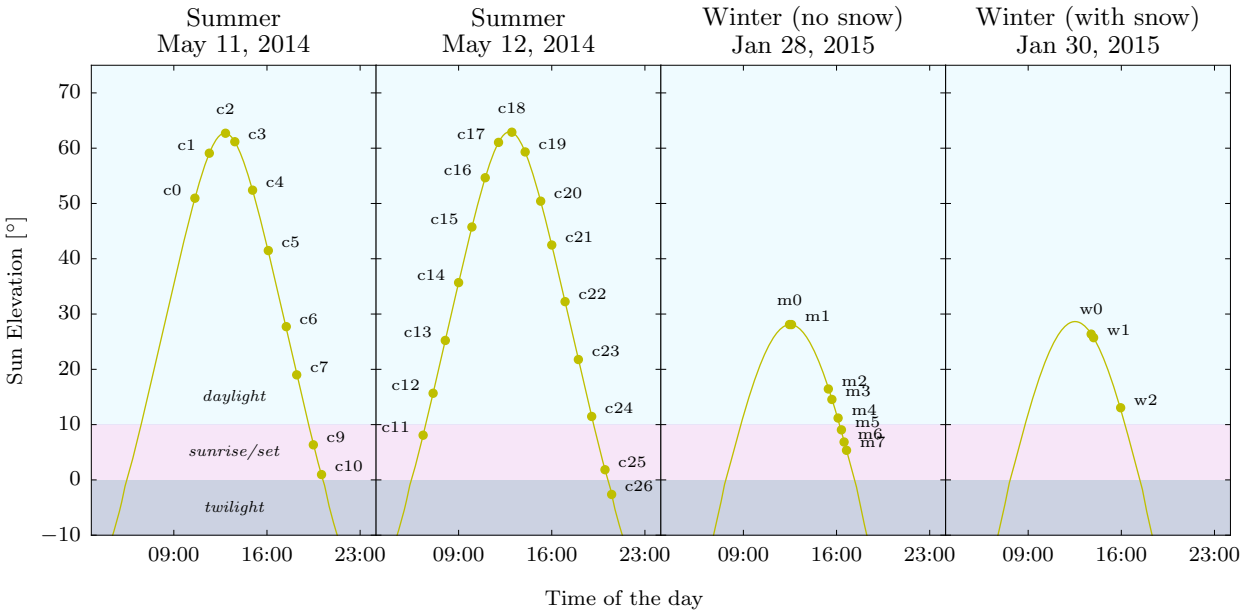
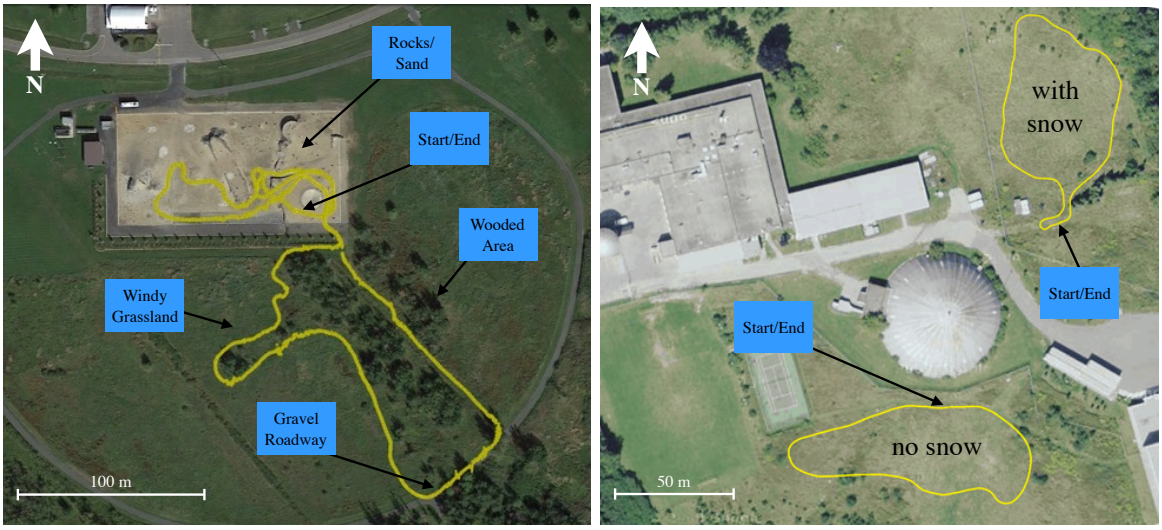


Figure 3.12: Overview of all recorded paths with respect to their time of the day and their sun elevation. The shaded areas correspond to different elevations where the luminosity significantly changes.



(a) Summer data set (45.518006, -73.393077). (b) Winter data sets (43.781970, -79.465030).

Figure 3.13: Annotated satellite imagery of the dynamic data sets recoded in different seasons. (a) Route recorded around the CSA's Mars Emulation Terrain. Two biomes of interest are present in this data set: Forest and Desert. (b) Routes recorded on UTIAS campus. The trajectory selected when there was no snow is at the bottom and the path with snow at the top of the image. Credit for the satellite imagery: Imagery ©2015 Google, Map data ©2015 Google.

significantly changes the exposure of the image. This field test proceeded by teaching an approximately 100 m loop, shown in Figure 3.13, through this environment. The path was taught when the sun was at its highest elevation point. The robot autonomously repeated the path seven times between 15:20 and 16:50, when the sun was setting (i.e., sunset happens much earlier during winter).

Winter (with snow) This field test was designed to test the multi-stereo system's ability to perform autonomous navigation in snowy environments. Snow is an especially difficult environment for vision-based systems as it is practically contrast free, causing a lack of visual keypoints in most of the scene. Snow cover changes shape quickly as well. It accumulates, melts, turns to ice, and can be blown by the wind changing the shape of the ground within minutes. Snow is also highly reflective; on sunny days this can lead a camera's autoexposure to generate images that are overexposed. An example of this environment can be seen in Figure 3.11c, where the Grizzly is traversing through a snow covered field. A 250 m path, shown in Figure 3.13, was manually driven through a large field with fresh snow cover as a teaching pass. During the teach, the sun was at its highest point in the sky, causing significant overexposure of the camera. The path was autonomously repeated approximately three hours later, when the elevation of the sun was significantly different. The complexity of the deployment and hardware limitations during this cold and windy day lead to a smaller number of repeats compared to the other data set. Nonetheless, it is enough to draw a comparison with other environments.

3.4.3 System Configuration

This section provides a brief overview of our system configuration and algorithm parameters since they can influence our results. More precisely, we detail the following steps of the localization pipeline: keypoint detection, keypoint matching, and outlier rejection. Details on the relevant parameters used can be found in Table 3.2.

Keypoint Creation Keypoints in all field tests were detected and described with upright SURF (Bay et al., 2008) using the GPU SURF library (Furgale and Tong, 2010). In this implementation, detected keypoints are binned to ensure a uniform distribution across the image.

Keypoint Matching Given a query keyframe, K_q, and a reference keyframe, k_r, our matching method seeks to find a match for every keypoint in k_r. Candidate matches are considered if the descriptor distance is below a threshold and they are within the current searching window, which filters out matches that are too far from the keypoint in pixel space. This window expands during localization failures for a more thorough search for candidate matches. For each keypoint in k_r, the candidate match with the highest matching score is selected.

Outlier Rejection Our outlier rejection method uses a simple RANSAC implementation that uses the Horn three-point method (Horn, 1987) as its model. Potential inliers are evaluated based on the reprojection error after being transformed to the candidate solution.

Table 3.2: Relevant parameters for the autonomous path-following field tests.

Parameter	Value
Minimum match count for localization	6
Maximum distance allowed on dead reckoning	20 m
Translation to add keyframe	0.2 m
Rotation to add keyframe	2.0 degrees
Target number of keypoints	600
RANSAC iterations	600
RANSAC inlier threshold	4.0 std. dev.
Matching search window (prior localization success)	11×8 pixels
Matching search window (prior localization failure)	133×100 pixels

3.5 Evaluation Metrics

To evaluate the impact of appearance change on visual navigation, we selected three quantitative metrics: Keypoint Quantity, Keypoint Sparsity, and Keypoint Quality.

Keypoint Quantity This is a notion of the number of total inlier matches observed at any point in time between the live keyframe and the map keyframe during an autonomous traverse. Over the course of a day, this number decreases; if it drops too low, the system will be forced to rely on VO, and eventually fail to localize relative to the taught path. If the system is unable to relocalize to the taught path, within a pre-specified distance, the system is set to stop. This metric is analyzed in Section 3.6.1 with respect to color-constant images, and in Section 3.6.2 with respect to multiple stereo cameras.

Keypoint Sparsity The keypoint quantity alone is an insufficient metric to ensure precise path following. During an autonomous traverse, keypoint matches to the map can be distributed unevenly through a given path. The previously mentioned metric of keypoint quantity aggregates data through a full repeat trajectory, limiting the analysis on consecutive successful localizations. We can indirectly observe the sparsity of keypoint matches by observing the distance the robot relied on VO before being able to localize to the map. A short distance driven while relying on VO is sign of a robust solution against the environment traversed. A system relying entirely on VO for a long period of time will increase its position uncertainty and will drift away from its reference trajectory leading to a mission failure. In our system, if the dead-reckoning (VO) distance is over 20 m, the system will stop and the run is considered a failure. This metric is analyzed in Section 3.6.1 with respect to color-constant images, and in Section 3.6.2 with respect to multiple stereo cameras.

Keypoint Quality Apart from quantity and sparsity, keypoint *quality* is equally important in judging the accuracy of localization. 3D landmarks measured with a stereo camera have depth uncertainty associated between the left and right keypoint matches. As this disparity decreases, the depth becomes more sensitive to these changes, and uncertainty associated with the depth reconstruction increases. High uncertainty is correlated to keypoints observed far from the camera (i.e., in the background of the image). A reliance on background keypoints will lead to an inaccurate translation estimate, providing

poor information to the path tracker and potentially endangering the vehicle. This metric is analyzed in Section 3.6.3 with respect to expected keypoint quality in varying environments.

3.6 Results

The goal of this section is to demonstrate significantly improved robustness to temporal and environmental change when the MCL-based systems are used. This is achieved through metric analysis covering hourly changes, weather changes, seasonal changes, and different biomes using the different evaluation metrics described in the previous section. An overview of the evaluated techniques is provided in Table 3.3.

Table 3.3: Overview of the solutions evaluated and compared using the MCL field testing data sets.

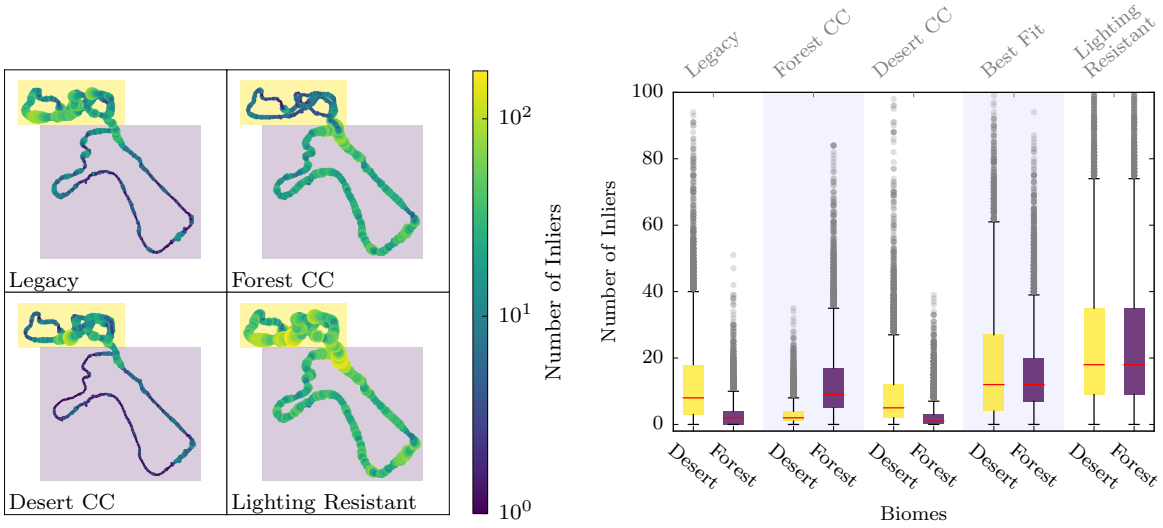
Solution Name	No. Channels	Precision
Legacy	1	Relies on the green channel (grayscale) of an image to extract keypoints as in (Furgale and Barfoot, 2010).
Forest-CC	1	Converts the RGB channels to a single Color-Constant channel with the parameters tuned for Forest biomes.
Desert-CC	1	Converts the RGB channels to a single Color-Constant channel with the parameters tuned for Desert biomes.
Best Fit	1	Selection of either the Legacy or Forest-CC channel based on the number of keypoints extracted as in (McManus et al., 2014a).
Lighting Resistant	3	Combination of Legacy, Forest-CC and Desert-CC channels.
Dual Legacy	2	Combination of the Legacy channel for the rear and front camera.
Dual Lighting Resistant	6	Combination of the Lighting-Resistant channel for the rear and front camera.

3.6.1 Color-Constant Images During Path Following

Results of the static experiments demonstrated a significant improvement in terms of an increase in feature matches for the different color-constant solutions (i.e., Forest CC and Desert CC). However, these experiments do not cover the performance of color-constant images when the viewpoint is not perfectly aligned, which is the typical case for autonomous path following. To test this sensitivity, we analyze the performance of our multi-channel VT&R system using the Summer data set, where the path was autonomously repeated 26 times. We use the results to validate our trained parameters, where other factors could influence the quantity of keypoints.

We first focus on the impact of the *Desert* and *Forest* biomes on different solutions in term of keypoint quantity. We use the single repeat run c4 (03 hours 10 minutes after the initial run) as a representative example. The full 1 km path was clustered in two groups, represented in Figure 3.14a as shaded areas. The color and size of the points in the figure give a qualitative representation of the behavior of the different solutions in each type of biome. We can observe that the lighting-resistant solution presents a larger stability over the full trajectory when compared to individual channels. A more quantitative evaluation is presented in Figure 3.14b, where Tukey boxplots are used to depict the medians and the interquartile ranges. These distributions give an idea of the expected number of inliers of an image in different biomes. We can observe that the Forest CC performs significantly better[1] in the Forest biome and inversely for the Desert CC solution. The lighting-resistant solution performs similarly across the

[1]We use the median outside the interquartile range of the compared distributions as a simple significance test.



(a) Qualitative results.

(b) Quantitative results.

Figure 3.14: Impact of different biomes (Desert and Forest) on the number of inlier matches for a single run 03 hours and 10 minutes after the original run. The quantity of inlier matches found is directly related to the stability of the localization system. (a) Maps representing the number of inliers at different locations on the path. The size and color of the points give a visual representation of the number of inliers at that location. The different biomes are represented by the shaded areas, with yellow and purple boxes representing desert and forest respectively. (b) Boxplots showing the distribution of inliers for different solutions and biomes. Results are paired by solution with the colors of the boxplots representing the two biomes under evaluation.

biomes demonstrating the complementary of the different channels when combined together. The Best Fit solution performs better compared to each individual channel (i.e., Legacy, Forest CC and Desert CC), but generates 33% fewer keypoints than the lighting-resistant solution due to its switching behavior. These results show that the core idea of the MCL algorithm—that combining data correspondences from multiple channels to solve a single state estimate—has a signficant impact on the performance of the system.

Results of Figure 3.14 only describe a single run. To analyze the stability of the number of keypoints through time, we computed the median value and interquartile range for all 26 runs of the Summer data set, independent of the specific biome. These results are displayed in Figure 3.15, with lines representing median values and shaded areas representing interquartile ranges. Results are divided between the first and second day, where weather conditions were mostly sunny and overcast, respectively. For clarity, only the results from the Legacy and lighting-resistant solutions are presented, but the lighting-resistant solution produces more inliers than all solutions presented in Figure 3.14 at all points in time. In Figure 3.15, we can observe two major points. There is a stark contrast between overcast and sunny days when deploying a vision-based path-following algorithm. On sunny days, the number of inliers sharply drops over the course of the day to due to the changing elevation, and then briefly rises after sunset. On overcast days, the sun elevation has less impact on the shadow positions producing a more constant number of matches through the day. The strength of the lighting-resistant system is most apparent on day one between 14:00 and 19:00. This time period corresponds to the setting sun, when the shadows grow very large in the opposite direction of the appearance of the map. During this time

period the lighting-resistant system maintains a low, but acceptable inlier count for the traverse, while the Legacy system drops to a median value of zero.

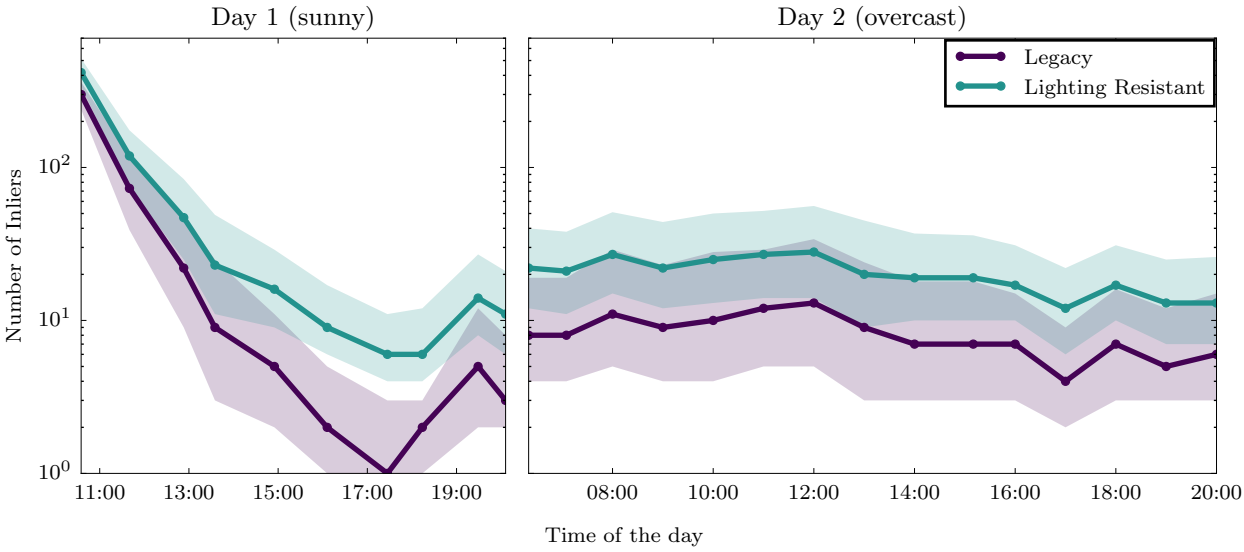
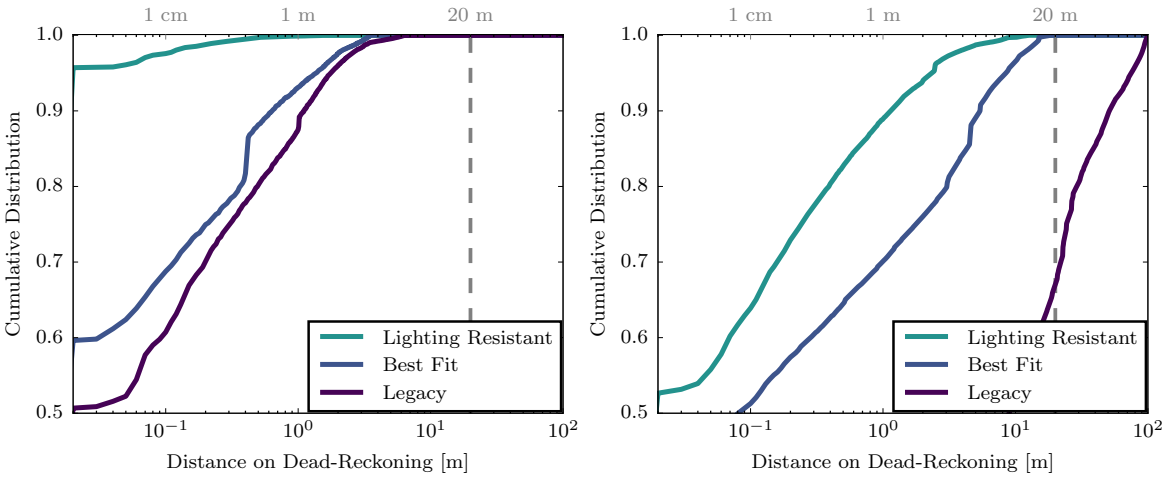


Figure 3.15: Evolution of the number of inlier matches through both experimental days in summer. The quantity of inlier matches found is directly related to the stability of the localization system. The advantage of the lighting-resistant solution is more significant during day one, when conditions were sunny. The lines correspond to the median number through the full 1 km-long path realized at that time of the day and the shaded area defines the interquartile distances of 25%-75%. Note log scale on the y-axis.

The number of inliers only present a partial view on the stability of a solution. The second part of our analysis focuses on keypoint sparsity, which can only be evaluated when a camera is moving. During an autonomous traversal, the robot attempts localization at the frame rate of the camera. If a localization attempt is unsuccessful, the robot will use the VO output for its state estimate. The keypoint sparsity metric measures how often this occurs. Figure 3.16 shows the cumulative fractional distances the vehicle traveled on dead reckoning before being able to find enough inlier matches between images from the original path and images from a repeat three hours (Figure 3.16a) and seven hours (Figure 3.16b) after map creation. For example, the dark purple line in Figure 3.16a represents the results of the legacy system and shows that for 10% of the traverse, the robot would have driven more than 1 m on dead reckoning. This distance increases to 50 m on dead reckoning in Figure 3.16b. Short distances on dead reckoning demonstrate stability through the full path and indicate a safer traverse. The graph shows that both the Lighting Resistant and Best Fit solutions maintain a dead-reckoning distance under our safety threshold of 20 m. The improvement of the lighting-resistant solution over Best Fit is more apparent later in the day (i.e., around 20% after 7 hours), where the light changes are more critical.

Keeping the same evaluation metric (i.e., keypoint sparsity), we investigate how the distance on dead reckoning evolves through time more specifically for the lighting-resistant solution. Figure 3.17 shows results for all 26 runs of the field test. When looking at the results of Day 1 (Figure 3.17a), we see a continuous degradation of performance through time except for three curves that stand out. During the run c5, which is represented with a dotted line, thin clouds were passing rapidly over the sun. This happened only during that run and significantly changed the temporal trend observed with the other



(a) Summer data set, 3 hours (c4) after map creation. (b) Summer data set, 7 hours (c7) after map creation.

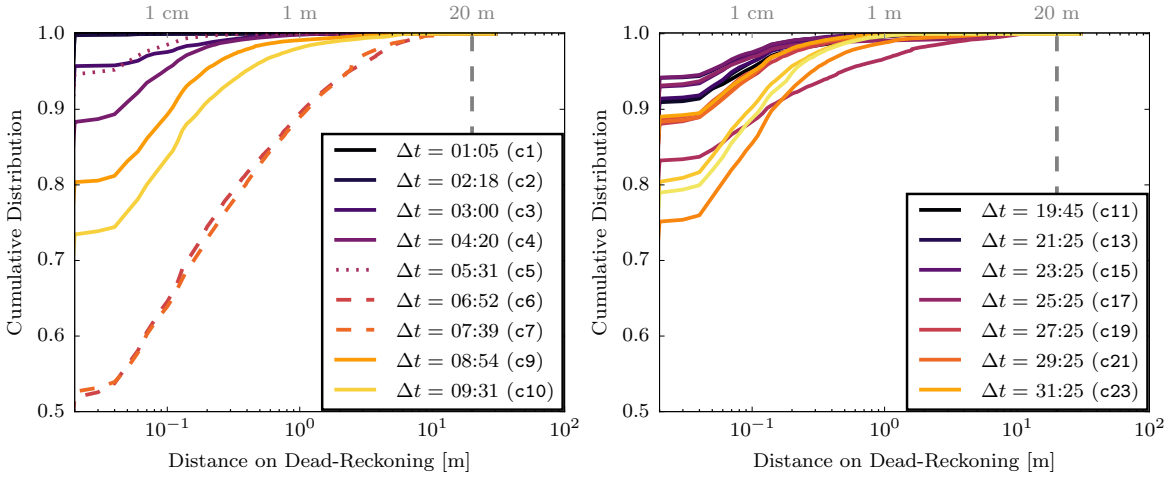
Figure 3.16: Comparative results between different localization systems based on the distance the vehicle would have had to travel on dead reckoning. A shorter distance indicates a more robust localization system. The dashed vertical gray line corresponds to a threshold where the autonomous drive is stopped for safety reasons leading to a mission failure. Note log scale on the x-axis.

runs, which were under a bright sun. When comparing c5 with the curves from the second day, which was overcast, we can observe a similar trend. The two curves with the worst performances (c6 and c7) occur just before sunset, where long shadows produced images that were very different in appearance from the map images. The runs c9 and c10 were captured after sunset, where the light is very similar to an overcast day. Figure 3.17b shows the 16 runs of day two all clustered in the same location with slightly better results for runs close to 24 hours after map creation. This shows again the positive impact of overcast days on vision-based path-following algorithms.

The last evaluations presented stable results through different weather conditions and biomes for the lighting-resistant solution. To push the analysis further, we investigated the impact of different seasons on the number of matches. Figure 3.18 shows the same solution (i.e., Lighting Resistant) for the Summer, Winter (no snow) and Winter (with snow) data sets. We can observe that the number of keypoints quickly reduces to a critical number of matches, reducing the temporal workspace of the solution. The next section investigates a proposed solution to cope with this situation.

3.6.2 Extended Field of View

One of the problems encountered in the Winter field testsre is that parts of the environment that rapidly change cause large areas with few to no keypoint matches. This is partly due to the fact that during winter, the sun is lower on the horizon for a longer period of time (when away from the equator, see Figure 3.12). This increases the chance of sun glare completely or partially saturating an image and accelerates lighting change due to quickly moving, long shadows. Furthermore, the presence of snow causes areas that are free of texture, which results in extremely sparse keypoint generation. With only small areas of the environment generating most of the keypoints, extending the field of view of the algorithm augments the chances of tracking a safe number of keypoints through the whole run. In this section, we investigate the impact of adding a rear camera to cope with these harsh seasons. Figure 3.19 shows the impact on the number of inlier matches for an autonomous traverse. The results show a large



(a) Summer data set, day 1.

(b) Summer data set, day 2.

Figure 3.17: Evolution of the distance traveled on dead reckoning over the course of two days for the lighting-resistant solution. A shorter distance indicates a more robust localization system. The different curves represent different runs spaced by roughly an hour. In the legend, Δt corresponds to the time separating the teach and the repeat path and in (b), only every second label is displayed for space consideration. Note log scale on the x-axis.

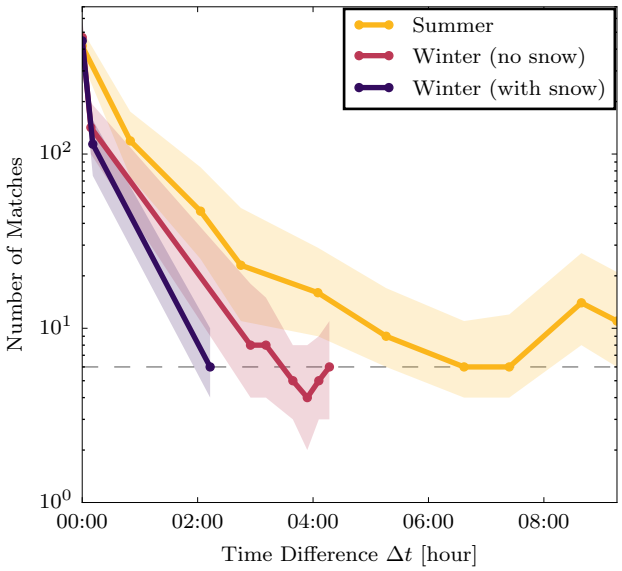


Figure 3.18: Evolution of the number of inlier matches for the lighting-resistant solution for multiple VT&R field tests spanning multiple seasons. All tests were conducted on sunny days when the advantages of the lighting-resistant solution are most apparent. The thick lines correspond to the median number through a full repeat path and the shaded area defines the interquartile distance 25%-75%. The x-axis represents the time difference, Δt, between the teach and the repeat path. Note log scale on the y-axis.

number of inliers being picked up by the front camera before moving to the rear camera. This handoff of keypoint matches between stereo cameras essentially doubles the amount of time that the stable feature in the environment is observed.

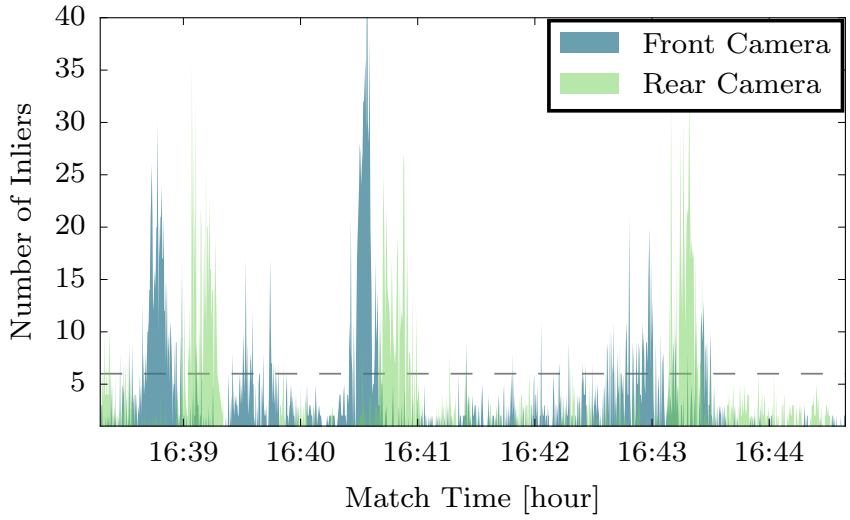
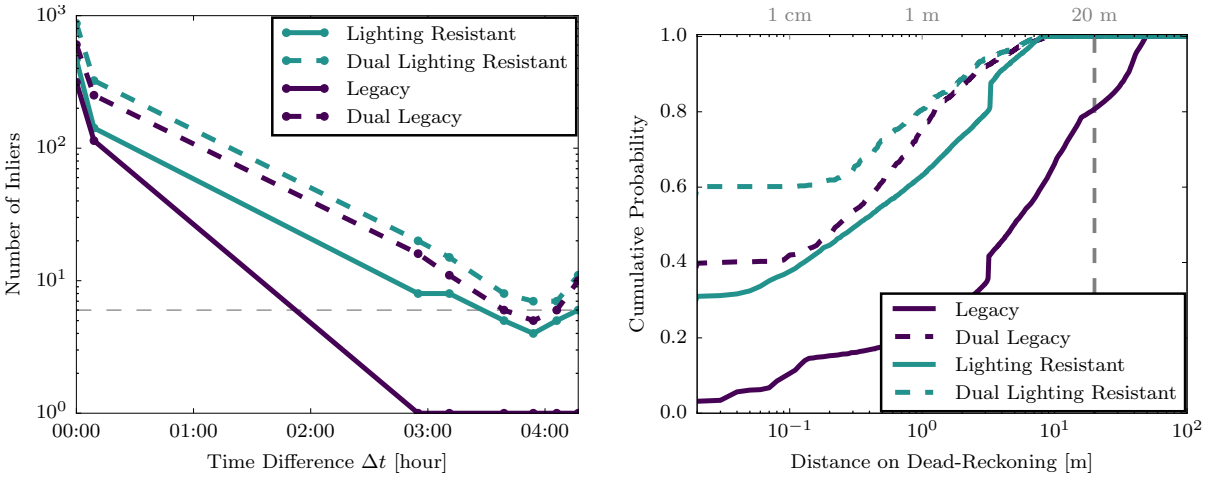


Figure 3.19: Evolution of the number of inliers for the front and rear cameras during a single repeat path. The dashed horizontal gray line corresponds to a safety threshold under which the the vehicle is not localizing against the teach image and moves on dead reckoning. A higher number of inlier matches indicate a more robust localization system. Combining inliers from both cameras significantly increases this robustness.

Using the combination of both cameras in parallel increases the chance of maintaining enough inlier matches to safely traverse these problematic environments. This ability is analyzed by investigating keypoint quantity in Figure 3.20a for the Winter (no snow) data set. We compare the legacy and lighting resistant solutions when using only the front camera (solid lines) and when using the front and rear camera (dashed lines). There is a failure case for the Legacy system using only the front camera near Δt = 02:55 (m2) because of the sun shining directly into the lens, which caused the system to completely lose track of its location. This repeat was not a failure case in the field because we were using the dual-camera solution. Also, we can observe that adding an extra camera to the system has a greater impact for the legacy system when compared to the lighting-resistant solution. The median number of inliers approach the critical threshold (dashed line) for the best solution at Δt = 03:54 where the high contrast of the images was mostly generating silhouettes on the horizon. We used this particular run to also look at the keypoint sparsity as depicted in Figure 3.20b. The Dual-Lighting-Resistant method still performs the best but the marginal performance improvement does not warrant the extra computation cost.

3.6.3 Keypoint Quality

This chapter presented evidence that seasonal changes, in particular the movement of the sun through the sky, accelerates the rate at which the number of keypoint matches decay through time (recall Figure 3.18). A second problem amplifying the difficulties of autonomous path-following algorithms in winter is that, on top of losing keypoints, the keypoints remaining tend to cluster on the horizon line. This phenomenon is illustrated in Figure 3.21, where vertical distributions of keypoint coordinates over the v-axis are presented for the three seasonal data sets. We can observe a rapid migration of the keypoints to the



(a) Keypoint quantity.

(b) Keypoint sparsity 03:54 hours after the map creation (m5).

Figure 3.20: Impact of adding a second camera on the number of inliers and dead-reckoning distance for the Winter (without snow) data set with respect to different solutions. (a) The number of inliers through time. More inlier matches correspond to a more stable localization system. Note log scale on the y-axis. (b) Evolution of the distance traveled on dead reckoning for multiple solutions. A shorter distance indicates a more robust localization system. The dashed vertical gray line corresponds to a safety threshold where the autonomous drive is stopped for safety issues. Note log scale on the x-axis.

top of the image for both Winter data sets. In all environments, it is expected that keypoints on the ground (i.e., lower in the v-axis of the image) will decay faster than higher points due to a number of reasons: (i) features seen here are observed at the centimeter level, while horizon features are observed at the meter level, (ii) shadows have a more pronounced effect on the ground plane near the camera, and (iii) terrain modification caused by the robot. The rapid decay of close matches in the winter can be attributed to accelerated lighting change, melting snow, and the high reflectivity of snow.

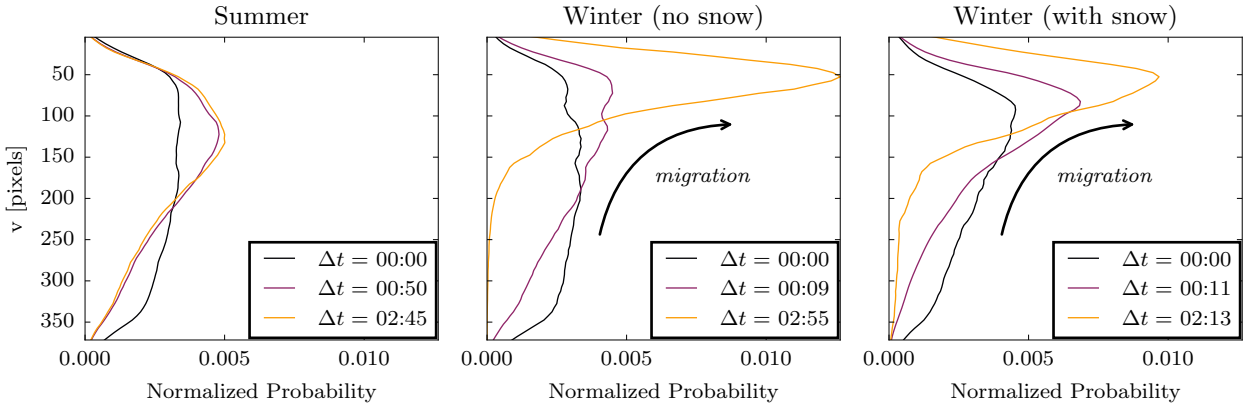


Figure 3.21: Vertical distribution of the matched inlier keypoints in the image coordinate frame. On the v-axis, zero corresponds to a keypoint at top of the image and 360 at the bottom of the image. All distributions are normalized and represented over a time period of several hours for different data sets.

This keypoint migration greatly impacts the accuracy of the localization system as keypoints with large depth uncertainty (i.e., at the horizon) reduce the accuracy of the translation estimation. The

impact of the keypoints moving up to the horizon line is explained with Figure 3.22, where the median and the interquartile distance of keypoint depths is plotted. The expected depth for the Winter data set increases to 42.7 m reducing the localization capability to that of a visual compass.

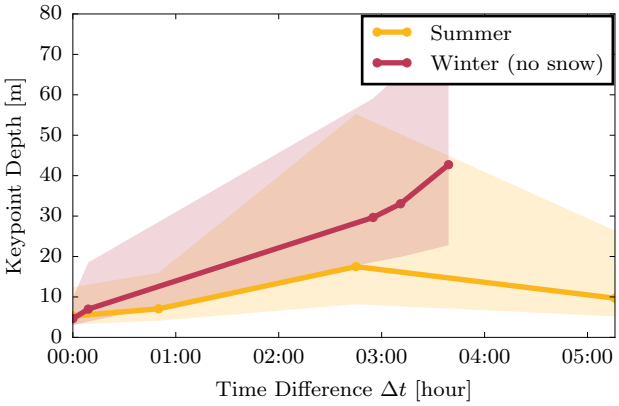


Figure 3.22: Comparison of the expected depth values in Summer and in Winter. The lines represent the median and the shaded areas the interquartile ranges of 25-75%. High depth values augment uncertainty on translation estimations.

3.7 Discussion

Our results have shown a significant improvement in robustness to temporal and environmental change when the MCL-based systems are used. In Section 3.3.3, using static timelapse imagery of specific environments, we were able to experimentally tune color-constant image transformations to achieve superior performance with respect to SURF keypoint matching. In Section 3.6.1, we used these color-constant image transformations to experimentally validate our lighting-resistant, multi-channel VT&R framework. We have shown a significant increase in localization performance across all analyzed metrics using our lighting-resistant method when compared to other closely related methods (Furgale and Barfoot, 2010; McManus et al., 2014a). Section 3.6.2 demonstrated a further increase in performance when multiple stereo cameras are used in a mutli-channel framework. By fusing data correspondences from multiple cameras into a single state estimation problem, we extend the field of view of the navigation system. Finally, in Section 3.6.3, we explored the impact seasons have on the performance of our navigation system. Our analysis of keypoint quality shows that winter environments are still challenging for appearance-based localization due to accelerated lighting change and a lack of contrast in the scene.

Multi-Channel Localization Our multi-channel localization system performs independent detection and tracking of keypoints for multiple information channels and combines matches from all channels into a single state estimation problem. This is an important distinction from the *Best Fit* method described in McManus et al. (2014a), where the full state estimation problem is performed in parallel for each channel and only the best result is used as the final estimate. We argue that combining keypoint matches from multiple channels into a single state estimation problem greatly increases the autonomy capabilities of a vision-based system when the appearance begins to change. In this case, the minimum number of required keypoints can be spread across all channels, allowing for localization in keypoint-limited environments. This is backed by our post-field analysis results (see Figure 3.14 and Figure 3.16)

Hourly Changes If the number of inlier matches drops too low, the system will be forced to rely on VO, and eventually will fail at following the taught trajectory. Figure 3.23 shows an illustration of the trend associated with the number of inlier matches typically observed over the course of a day. This figure sums up the experience collected over all of the field trials. On overcast days, there is a gradual decline in keypoint matches, because the appearance of the scene is generally constant. This is not true on sunny days, where an early drop is caused by the sun changing position and creating sharp, moving shadows on the ground. Keypoint quantity begins to rise again at the beginning of twilight, when the light from the sun is not directly observable, generating a shadow-less environment similar to an overcast day. The duration and time of sunrise, sunset, and twilight are dependent on the environment. For example, if the robot is in a canyon-like environment, the sun may disappear faster than usual. As a result, modeling the correlation between factors such as sun elevation and keypoint quantity is a non-trivial task.

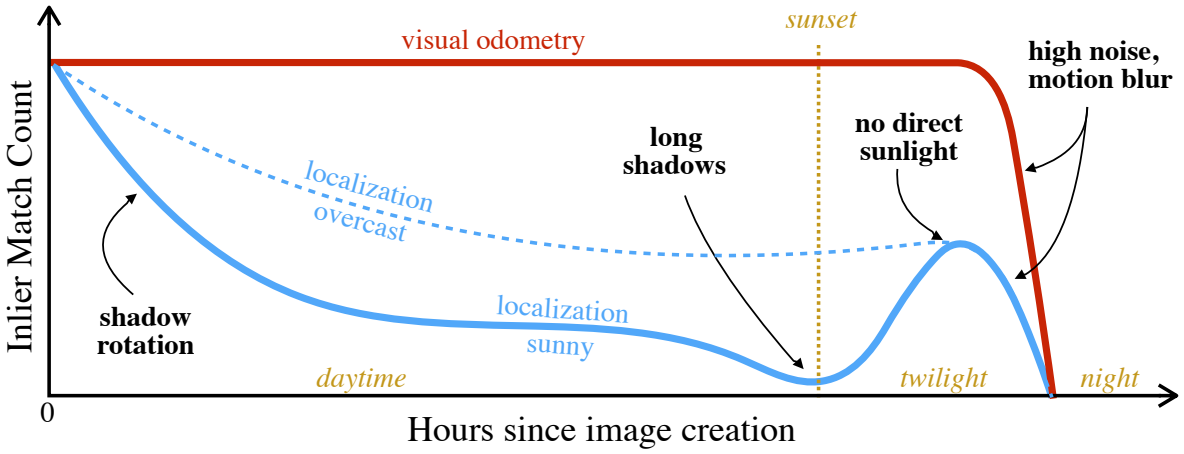


Figure 3.23: Illustration of the evolution of the number of inlier matches through a nominal day. Time zero corresponds to when the reference images are collected (teaching phase) and the blue line represents the typical slow degradation of the number of matches when matching current images to the teaching phase. The difference between a sunny day (solid line) and an overcast day (dashed line) is also included. The red line represents the number of keypoints used during VO, which stays constant up to the limit of the sensor. Yellow annotations refer to time events and black annotations refer to the main causes of inlier decreases or increases.

Snow During the teaching phase of the Winter (with snow) data set, it was bright and sunny. Due to the high reflectivity of the snow, this caused unforeseen issues for our stereo cameras. The brightness of the scene brought the factory settings of the autoexposure algorithm of the PGR Bumblebee XB3 to the limit. The result was saturated images, which reduced details in the foreground. This issue can potentially be overcome by overriding the camera's exposure limits or by manual control of the camera's exposure settings. The Winter (with snow) data set was collected when there was light snow cover. We also attempted to perform autonomous path following in deep snow conditions with unsatisfactory results (see Figure 3.24). In light snow, small vegetation is often visible in the foreground, providing visual keypoints with high contrast. In deep snow, these keypoints are gone and what remains in the foreground is nearly featureless. The only usable matched keypoints were on the horizon not only for localization, but also for VO. This caused frequent inaccurate pose estimates, which caused issues for

the path tracker. The problem of keypoint migration explained in Section 3.6.3 is even more apparent when a large quantity of snow is present.

Figure 3.24: Photograph from an attempted autonomous traversal in the deep snow. A lack of visual keypoints in the foreground resulted in poor localization and VO estimates.

Glare An initial hypothesis motivating the dual-camera field deployments was the assumption that the low elevation of the sun would cause glare in the camera, making localization impossible. Due in part to the attitude of the stereo cameras, glare was not the main issue. With the cameras tilted to the ground by 20 degrees, the sun was in the worst case only at the top of the image. Furthermore, we observed cases where sun glare increased the contrast of horizon keypoints, providing a significant boost in keypoint count. This has an indirect impact on the keypoint quality, but did not completely blind the camera. However, glare would be an issue if the cameras were pointed at the horizon.

(a) Balanced exposure.

(b) Over exposure causing high saturation.

(c) Wrong white balance causing blue-color shift.

Figure 3.25: Images extracted from a static data set recorded in a snow covered environment. The auto-settings of the PGR Bumblebee XB3 is causing artefacts in most of the images limiting our interpretation of the calibration results in snow.

Color-Constancy in Winter The color-constant image transformations are designed to remove the effects of lighting from an image. These were used to great success in the Summer field trials. In these trials, the robot repeated a 1 km route 26 times with an autonomy rate of 99.9% of distance traveled in

nearly every daylight condition. With this prior knowledge, the color transformations were expected to boost performance in the winter field trials as well, but this was not the case. A hypothesis is that the color-constant images were tuned to perform in green vegetation and red-rocks-and-sand. Further investigation was performed to experimentally tune a color-constant transformation for snowy environments using the techniques described in Section 3.3.3. Results from the experiment were inconclusive, with the experimentally found Snow-CC transformation underperforming compared to traditional grayscale images. This is primarily due to poor testing conditions which are displayed in Figure 3.25. Ideally, images have a balanced exposure, as seen in Figure 3.25a. Unfortunately, the majority of the images captured during the experiment were either over exposed (Figure 3.25b) or were incorrectly white balanced (Figure 3.25c). Because we were using the automatic settings of the PGR Bumblebee XB3, which performed poorly, our confidence in the results of the experiment is low and thus not reported here.

3.8 Summary and Novel Contributions

This chapter presented Multi-Channel Localization (MCL): a localization and mapping algorithm that takes advantage of multiple channels of information to aid localization across appearance change. A key contribution of this algorithm is that landmarks independently tracked in all channels can be used to solve a single state estimation problem. We presented two instances of this algorithm, the first used color-constant images to increase resistance against lighting change and the second used multiple stereo cameras to extend the algorithm's field of view. With a series of field tests, we have shown that, through use of our multi-channel localization scheme, we are able to effectively extend the autonomy rate of single-experience, vision-based path-following systems from a few hours to multiple days in realistic, outdoor environments. We furthermore explored the effects of lighting change in multiple seasons and quantified their influences on our localization system. We experimentally validated this work through a series of field tests covering over 28 km of autonomous driving in difficult outdoor environments in varying seasons. The lighting resistant system was first published in the 2015 proceedings of the International Conference on Robotics and Automation (ICRA) (Paton et al., 2015a), the multi-stereo system was first published in the 2015 proceedings of the Canadian Conference on Computer and Robot Vision (CRV) (Paton et al., 2015b), performance of both systems with respect to winter environments was published in the proceedings of the International Conference on Field and Service Robotics (FSR) (Paton et al., 2015c), and the MCL framework was formalized and summarized in the 2017 special edition on Field and Service Robots of the Journal of Field Robotics (JFR) (Paton et al., 2017b).

In summary, the novel contributions of this early work are:

1. A multi-channel localization framework that performs independent tracking of point-based visual features for multiple information channels and fuses data correspondences from all channels into a single state estimation problem.

2. A lighting resistant localization system that uses the multi-channel framework to fuse data correspondences from grayscale images and color-constant images to improve performance across lighting change.

3. A multi-stereo localization system that uses the multi-channel framework to fuse data correspondences from multiple stereo cameras to increase the field of view of the localization system and improve performance across general appearance change.

4. An in-depth analysis of expected localization performance in varying seasons with insight on the limitations of single-experience localization systems that rely on point-based visual features in difficult winter environments.

5. A methodology to experimentally tune the color-constant image transformations to improve performance in a given environment with respect to visual features tracked across lighting change.

The results of this chapter show that the autonomy window of autonomous path-following algorithms can be extended from hours to days in unstructured, outdoor environments through multi-channel localization. However, industrial applications will require vision-based autonomous path-following methods to reliably operate over longer timescales where seasonal appearance change is a factor. In the next chapter we address this issue by adapting the many-to-one localization concepts presented here into a novel, multi-experience localization and mapping framework.

Chapter 4

Multi-Experience Localization

In Chapter 3, we presented Multi-Channel Localization (MCL): a localization and mapping framework designed to extend the operational window of autonomous path following from a few hours to multiple days in difficult outdoor conditions despite significant short-term appearance change. In this chapter, we present Multi-Experience Localization (MEL), a localization and mapping framework designed to provide metric localization across long-term appearance change, enabling long-term autonomous path-following systems.

4.1 Introduction

Long-term autonomy is a crucial requirement for path-following systems to be realistically usable in industrial applications. Examples include applications that consist of repeated traversals over constrained paths, such as factory floors, orchards, and mines and applications that consist of autonomous exploration and retrotraverse such as search-and-rescue and hazardous-exploration robots. A long-term autonomous path-following system usable by these applications will require metric localization with respect to the privileged, manually taught path as the appearance of the environment changes due to weather and seasonal effects such as snowfall, vegetation growth, and construction. This is a challenging task for vision-based systems that rely on associating the appearance of local point-based visual features between the live experience and the privileged (manual) experience to achieve metric localization. A viable solution to overcoming this issue is the use of multiple experiences in the map, with the intuition that the appearance gap between the live and privileged experience can be bridged using experiences gathered during previous traversals. This multi-experience concept was first introduced by Churchill and Newman (2013) as Experience-Based Navigation (EBN), where new experiences are added to the map as localization fails, and a set of parallel localizers are deployed online to find an experience that can be localized to the live experience. This work demonstrated two fundamental concepts of multi-experience systems: i) metric localization with point-based visual features can be achieved despite extreme appearance change using multiple experiences, and ii) long-term autonomy can be achieved with multi-experience systems if the appearance change is sufficiently captured in the map.

In this chapter, we present the primary contribution of this thesis, the Multi-Experience Localization (MEL) algorithm. This algorithm expands upon the ideas of EBN and incorporates the lighting-resistant system presented in Chapter 3 with the distinct goal of enabling long-term, vision-in-the-loop

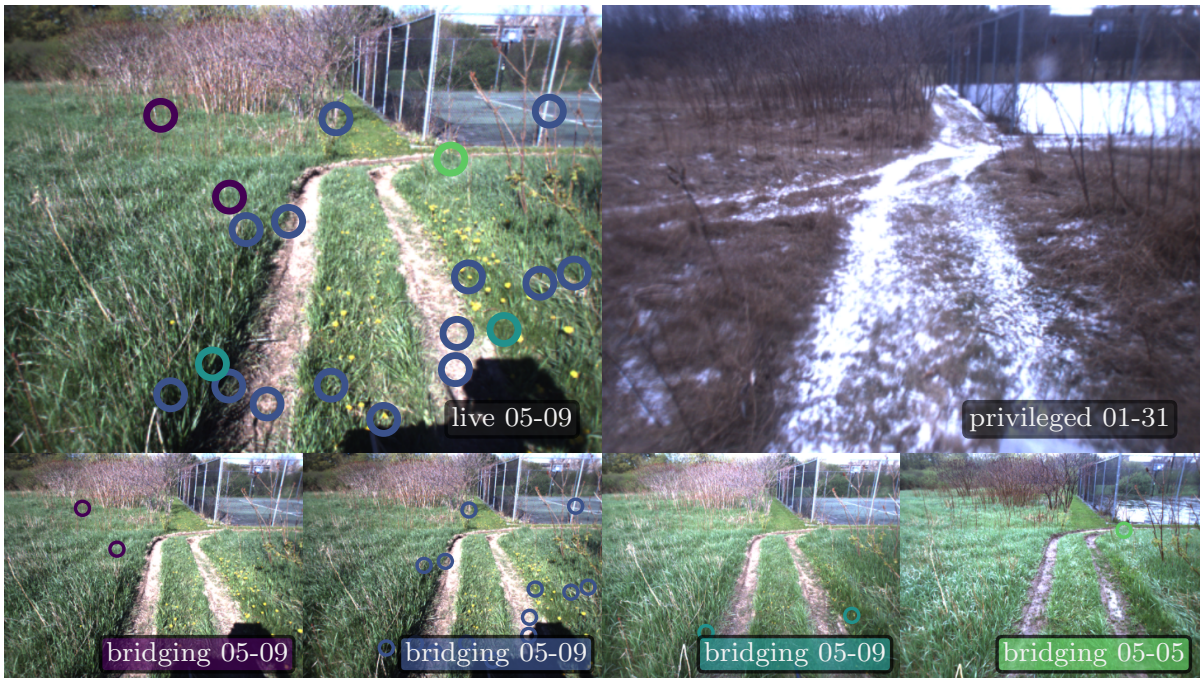


Figure 4.1: Illustration of the Multi-Experience Localization (MEL) algorithm. This metric localization algorithm designed specifically for autonomous path following estimates the pose of a live experience with respect to a manually taught privileged experience with uncertainty using intermediate experiences to *bridge the appearance gap*. Above, the live experience, captured on 05-09-17, has an inadequate number of feature matches to the privileged experience, captured on 01-31-17 to adequately localize. Bridging experiences metrically linked to the privileged experience and captured autonomously between 05-04-17 and 05-09-17 are matched to the live experience, allowing robust localization to the privileged experience despite extreme seasonal appearance change.

autonomous path following. This is accomplished by continuously estimating, with uncertainty, the localization between the live experience and a privileged (manual) experience using intermediate experiences simultaneously to bridge the appearance gap. An example of the algorithm can be seen in Figure 4.1.

To demonstrate the capability of the MEL algorithm, we conducted three unique experiments. The first experiment demonstrates the core concepts of the MEL algorithm: i) data associations between the live view and multiple bridging experiences can be used simultaneously to solve a single state estimation problem, and ii) the appearance gap between the live view and privileged view can be sufficiently bridged in real time using a fixed subset of bridging experiences. This offline performance analysis is conducted on a 9 km subset of the challenging 26 km CSA dataset detailed in Chapter 3, which exhibits significant appearance change due to lighting variation. The second experiment compares the performance of the MEL algorithm to its most related work, EBN (Churchill and Newman, 2013) on a challenging winter data set detailed in Section 5.3.4 that contains extreme appearance change due to snow fall and melt. The third experiment addresses concerns of the MEL algorithm's ability to minimize localization drift as the appearance of the scene changes. This "photocopy-of-a-photocopy" issue arises from the fact that the MEL algorithm will at some point only match to landmarks from bridging experiences, whose metric relation to the privileged experience is computed from previous, uncertain transformations. This online experiment consists of teaching a small, 50 m straight-line path and autonomously repeating the path back and forth over 180 times while collecting ground truth data with a Leica TotalStation. We show that even after 180 autonomous traversals, the accuracy of the VT&R 2.0 system with respect to the original path remains in the centimeter level.

The novel contributions of this chapter are: i) a data structure that relates multiple experiences together metrically, ii) a methodology to metrically localize a live experience to a privileged, manually driven experience using several intermediate experiences gathered during autonomous operation, iii) a methodology to bookkeep uncertainties in the multi-experience localizer, accounting for uncertain map landmarks originating from multiple experiences, and iv) experimental evaluations of the MEL system to validate the core ideas of metric localization using many experiences. The contributions in this chapter have appeared in the proceedings of the International Conference on Intelligent Robots and Systems (IROS) (Paton et al., 2016).

4.2 Related Work

This chapter presents the MEL algorithm, which enables metric localization across extreme appearance change through the use of a novel, multi-experience localization and mapping framework. This algorithm is designed with the specific use case of providing navigation for long-term autonomous path-following systems. In Section 3.2, we presented work related to autonomous path-following systems, either providing short-term autonomy through vision sensors, or long-term autonomy through active sensors. To the knowledge of this author, there is no work published on full vision-in-the-loop path-following systems capable of autonomous operation over seasonal changes using vision. As such, work related to this chapter is focused on inter-seasonal localization and is broken into topological and metric localization techniques.

4.2.1 Long-Term Topological Localization

There has been significant recent work on long-term topological localization, specifically addressing localization across seasonal appearance change. A notable advancement in this topic is localization through the alignment of image sequences. The task of alignment-based localization is to topologically localize a sequence of live input images to the best matching image sequence in a map through whole-image similarity matching. First introduced by Milford and Wyeth (2012) as SeqSLAM[1], this image-alignment method first computes a confusion matrix between an array of live images and the array of map images using whole image matching and then searches for the best diagonal path through the matrix. While effective at localizing across appearance change as extreme as night vs. day, this method assumes a constant velocity of the robot throughout the sequence and is highly susceptible to scale and viewpoint changes (Sunderhauf et al., 2013). In Pepperell et al. (2015), SeqSLAM was improved to handle viewpoint changes through automatic image scaling. Alignment-based localization was further refined by Naseer et al. (2014) where sequence alignment is formulated as a graph search problem, allowing for non-constant velocity and loop closures in the traverse. Improvements to this method were made through the use of global image features built from Deep Convolutional Neural Network (DCNN)s (Krizhevsky et al., 2012) in Naseer et al. (2015) to further improve localization across appearance change. Long-term topological localization can be further improved by learning image change prediction transformations. The key intuition behind these techniques is that most outdoor appearance change such as shadows from lighting and seasonal changes such as vegetation growth and snowfall is predictable and repeatable.

[1]Despite the name, SeqSLAM does not perform mapping and is limited to performing localization to a pre-existing map unlike topological SLAM methods (Cummins and Newman, 2008) that are able to detect when an input image is a new place.

Given examples of a scene's varying appearance, these techniques seek to learn a transformation that changes an input image's appearance to the that of a map image. Neubert et al. (2013) learn a dictionary that translates a vocabulary of visual words between appearances, but require pixel-level alignment of training sequences. Lowry and Milford (2016) use supervised linear regression to predict appearance change and unsupervised principal component analysis to apply change removal. These methods have the potential to greatly increase the performance of topological localization across extreme appearance change such as spring vs. winter.

While these methods are capable of reliable localization across extreme appearance change, topological methods on their own are not adequate for vision-in-the-loop navigation, often assuming low-level lane following or other control algorithms will provide metric localization to a path-tracking controller.

4.2.2 Long-Term Metric Localization

Computing vision-based, metric localization across appearance change is a prerequisite for long-term autonomous path-following systems that rely on passive sensors. However, this task is difficult due to the reliance of most metric-estimation methods on point-based visual features, illumination values, or gradient-constancy assumptions, which are all highly susceptible to appearance change.

Early Examples

Biber and Duckett (2005) represent the environment by a set of parallel maps that operate at separate timescales. Using these maps, ephemeral objects such as people can be picked up and tracked by a high-dynamic timescale map and treated as outliers in a low-dynamic timescale map. This can be thought as the difference between a short-term memory map and a long-term memory map. CAT-Graph+, developed by Lowry et al. (2012), extends the CAT-SLAM algorithm (Maddern et al., 2012b,a) by running a parallel, odometry-only particle filter to bridge the gap when the vision system fails due to a changed environment. In CAT-Graph, localization occurs by updating the particles via the robot's odometry and determining loop closures with a place recognition algorithm. When there are small perceptual changes in the environment, such as moved furniture, this system will fail. By running a parallel odometry particle filter, CAT-Graph+ can maintain a correct estimate until the visual system returns to a recognizable area. While these methods are effective, they rely on single-coordinate-frame maps best suited for small indoor environments where the majority of the scene stays the same. Autonomous path-following applications that operate outdoors will require large-scale operation in environments where the global appearance changes.

Dense Methods

Vision-based localization is susceptible to appearance change primarily due to the descriptors of sparse visual features whose appearances tend to rapidly change in outdoor environments. Intuitively, dense localization and mapping methods, ones that use the information encoded in the entire image to match and compute metric localization are more robust to appearance change. Wolcott and Eustice (2014) use dense techniques to localize a monocular image to a 3D prior map built from LiDAR scans. To localize, a grid of synthetic views are generated using the prior map, with each view representing the prior map warped and projected into the monocular camera frame by a specific pose. The camera pose is estimated by maximizing the normalized mutual information between the live monocular image and this grid of synthetic views. The method was further refined by Wolcott and Eustice (2015) to improve the map

representation and incorporate an Extended Kalman Filter (EKF) in the localization scheme. Pascoe et al. (2015b) build a prior map consisting of a 3D LiDAR mesh textured by appearance data from a camera. They localize monocular images to the prior map by minimizing the Normalized Information Distance (NID) between the monocular image and the prior map warped and projected into the camera frame by solving an iterative, nonlinear optimization problem.

The motivation behind this mono-image to lidar-map localization scheme is that a single vehicle with an expensive sensor can generate the prior map and a fleet of vehicles with inexpensive vision sensors can be deployed with it. While this is well suited for applications such as autonomous cars, it is less suited to single-vehicle applications and on-the-fly applications. The method introduced by Pascoe et al. (2015a) was formatted to work with only vision sensors in Pascoe et al. (2017). In this new formulation, the 3D prior map is built from monocular images using the Large Scale Direct (LSD) monocular Simultaneous Localization and Mapping (SLAM) framework (Engel et al., 2014). This method has the benefits of robustness to appearance change that dense localization methods offer without the reliance on an expensive active sensor. They demonstrate localization in varying appearances such as overcast, sunny, raining, night, and snow with varying success. While this method succeed at localizing across some appearance changes, it failed to localize at night and in the rain, and had limited success in sunshine and snow.

Sparse Methods

State estimation techniques that use point-based visual features benefit from low computation costs and sub-pixel measurements, allowing for fast and precise metric localization if data associations can be established between the live view and the map. However, traditional point-based visual features such as SURF (Bay et al., 2008) are highly susceptible to appearance change. As the appearance of the scene changes, so do the keypoint locations and descriptors of the features. Single-experience localization methods that rely on associating these features between the live view and a single static map quickly fail in outdoor environments, as demonstrated in Chapter 2 and Chapter 3.

Learned Visual Features One way of overcoming the issues of visual feature association across appearance change is to design features that are more robust to appearance change. Recent work into this task has been focused on using machine learning to train better visual features. McManus et al. (2015) present Scene Signatures: learned, place-dependent visual features that are robust to appearance change. Given a prior database of the scene under varying appearances, custom Support Vector Machine (SVM) classifiers that describe rectangular regions of the scene are trained to fire on a specific point across all trained appearances. Their method was shown to be able to perform metric localization across seasonal appearance change in urban environments, including rain and light snowfall. This method was further refined by Linegar et al. (2016) to use an unsupervised learning method, requiring only a single mapping example to train the classifiers using a series of novel tests to find areas of the image that are robust for localization. While proven to be effective at localization across appearance change in urban environments, these scene signatures provide "weak" localization, lacking the metric precision of traditional point-based visual features and requiring additional information such as assumptions about the environment and motion of the vehicle.

Multi-Experience Maps The strategy used by this thesis to enable long-term autonomy using sparse, feature-based methods is to build multi-experience maps that capture the appearance change of the scene. The intuition behind these methods is that the appearance of outdoor scenes change gradually and will often have a finite number of appearances. This can be exploited in sparse methods, where the output of the VO estimator is often the map. In these methods, whenever the robot traverses its environment this VO output can be added to the multi-experience map as a new experience of the environment, naturally capturing the appearance change of the scene as the robot continually operates. By providing many faces of the environment, this multi-experience map enables long-term localization across extreme appearance change. Interesting research questions arise related to how this multi-experience formulation can be feasibly used in a localization and mapping framework: i) how are the multiple experiences represented in the map?, ii) how do the multiple experiences relate to one another?, iii) how does the live view localize against a map containing many appearances of the same place?, iv) how does the algorithm remain computationally tractable as the number of experiences increases?, and v) how many experiences are required to capture the appearance change of a scene? It is in how these questions are answered where the differences in related work arise.

An early example of a data structure that makes use of multiple experiences is the view-based maps system presented by Konolige and Bowman (2009). In their work, an initial keyframe-based metric map in a global coordinate frame is built from the output of the stereo VO system. They interleave VO with an opportunistic place recognition algorithm that constantly searches the graph for loop closures. When place recognition to the map begins to fail, new keyframes from VO are added to the map. Upon successful loop closure, a global optimization problem is run to relax the graph. They prune their graph by keeping only a fixed amount of keyframes in the map, favored by appearance diversity. While this method is successful at minimal appearance change in indoor environments, it is not suited for large-scale outdoor operation under global appearance change.

The multi-experience map concept most closely related to this thesis was introduced by Churchill and Newman (2013) as the seminal Experience-Based Navigation (EBN) framework. As the Multi-Experience Localization (MEL) algorithm presented in this thesis is heavily inspired by this work, we present a detailed overview of the EBN framework here. The EBN framework is based on the multi-experience data structure shown in Figure 4.2. This data structure consists of a graph $G = \{V, E_m, E_t\}$, containing vertices (triangles) connected by metric and topological edges. Vertices, each with a reference frame, $\underrightarrow{\mathcal{F}}$, store raw sensor observations and triangulated 3D landmarks with associated descriptors. Metric edges (blue lines) in the graph link two vertices with a relative $SE(3)$ transformation, and topological links (dashed, orange lines) simply denote that two vertices are representing the same space. Experiences in the graph are sets of vertices connected by metric transformations, and can be thought of as the output of a VO system.

In this system, the map is built by storing new experiences when localization fails or is poor. Upon localization failure, the live VO output is saved to the map until localization to the map is once again established. This scheme of storing experiences builds a map where its sparseness in a specific section correlates to the amount of variance in the appearance of the scene, creating a naturally sparse map. Experiences are connected to each other through topological links, which are used during localization to initialize localizers. Topological links can be created in a few ways. At the start of a new experience, a link is made to the most recently localized experiences. If the live view successfully localizes to more than one experience, then links between these experiences are created. Links can also be created offline

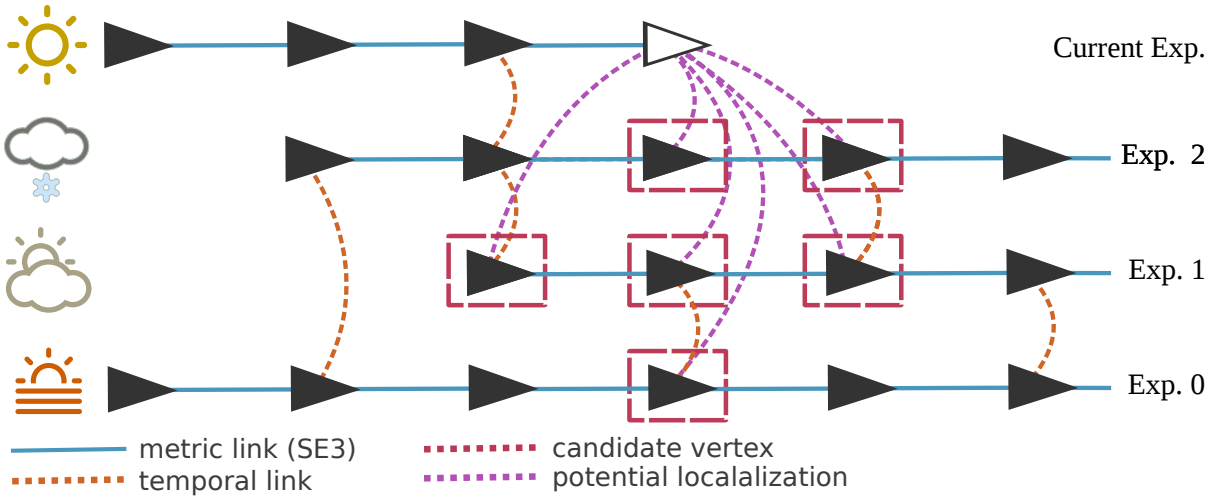


Figure 4.2: Overview of the Experience-Based Navigation (EBN) framework. This system is based on a multi-experience graph structure consisting of vertices (black triangles) connected by relative, $SE(3)$ transformations (blue lines) and topological edges (orange lines). An experience in the graph is defined as a set of vertices connected by metric edges and can be thought of as a pose graph generated by VO. When localization to the map is poor, the VO output of the live run is added as a new experience in the graph and can be related to prior experiences through topological links. To localize the live vertex (white triangle) against the map, a set of N parallel localizers (red, dashed squares) are run, resulting in N localizations, where the best result is accepted as the state estimate. The primary intuition behind EBN is that there is a finite number of appearances to a place and the graph structure should eventually stabilize to a fixed size.

through a place recognition algorithm such as FAB-Map (Cummins and Newman, 2008), or through the use of GPS coordinates stored at the vertices. The EBN system localizes across appearance change with this multi-experience map structure through the use of a collection of parallel, independent localizers. Each localizer (red, dashed rectangle) performs an independent state estimation problem, solving the metric transformation (purple, dashed line) from the live vertex (white triangle) to the map vertex (black triangle) within the square. At the end of localization, the system chooses the "best" result from the localizers as its state estimate. Topological links in the map are used to establish which localizers will be used in the state estimation problem.

If the EBN system is unbounded, i.e., if there are localizers used for *every* experience in the map, it will become computationally intractable as the number of experiences increases. While adding experiences only when the localizer fails reduces the number of experiences in the map, this is not the case for environments with highly variable appearance change which are typical in unstructured outdoor environments. This issue was addressed in Linegar et al. (2015), where past performance metrics are used to decide which experiences to use. In this scheme, the top N experiences most likely to be successful in localizing to the live view can be recommended for use in localization. This effectively caps the computation cost of the localization algorithm and allows for operation on much larger maps over longer time periods. EBN was further expanded through the use of active sensors in Maddern et al. (2015).

While EBN has proven highly effective at providing metric localization across seasonal changes, it is not readily suitable as the navigation component of an autonomous path-following system. This is because the resulting metric localization may be provided with respect to any of the prior experiences,

implying that each experience must be driven manually (or somehow labeled as 'safe' for autonomous repeating). In VT&R, there is always some path-tracking error (in addition to localization error); if we simply add new, independent experiences during autonomous operations, the path-tracking error will build from one experience to the next as the appearance shifts, analogously to the effect of taking a photocopy of a photocopy. To be able to continue to use a single privileged (manual) experience in VT&R, we must use several experiences simultaneously, rather than independently, in order to localize safely. Furthermore, a welcome side effect of simultaneously localizing to many experiences is an increase in the amount of inler matches in the state estimation problem, especially when the appearance of the live experience is significantly different. These are the core ideas behind the MEL algorithm presented in this section and can be viewed as a generalization of EBN to support VT&R. Because of this similarity, we conducted a detailed performance comparison of the two systems to quantify the differences in both localizers. The experimental setup and quantitative analysis of this comparison can be found in Section 4.4.2 and Section 4.6.2, respectively.

Apart from EBN, another example of a sparse, multi-experience localization system is the "Summary Maps" framework, presented in Muhlfellner et al. (2015). This method provides accurate, metric localization across seasonal appearance change through a multi-experience map that is pruned and curated offline. This mapping strategy was validated through a multi-season data set where different offline maintenance techniques were compared to ensure real-time performance for a large number of experiences. Online localization using Summary Maps has been shown to provide accurate, metric localization across seasonal appearance change. While successful, this method requires downtime between traverses to perform mapping on an offline server, which is not ideal for applications that require constant operation or quick turn around between manual demonstration and traversals.

4.3 Methodology

The MEL system outlined in this section is designed to enable robust localization for long-term, autonomous path following through a novel multi-experience localization and mapping framework. This section begins with the data structure used to represent the multi-experience map, then details our stereo VO pipeline used to generate experiences and update the robot's state, and finally the MEL algorithm that provides long-term metric localization.

4.3.1 The Spatio-Temporal Pose Graph

The MEL system uses the Spatio-Temporal Pose Graph (STPG) data structure as its multi-experience map. This data structure allows the MEL system to: i) differentiate between manually and autonomously driven experiences, ii) relate autonomous experiences to manually driven experiences metrically with relative, uncertain $SE(3)$ transformations, iii) store arbitrary data structures with random access in each experience, and iv) represent a network of driveable, connected paths suitable for use in route planning algorithms.

Depicted in Figure 4.3, this graph structure contains vertices, temporal edges, and spatial edges. Vertices, each with a reference frame, $\underrightarrow{\mathcal{F}}$, store raw sensor observations and triangulated 3D landmarks with associated covariances and descriptors for each information channel. Landmarks from multiple channels in this data structure are stored in the map indentically to the MCL method described in Section 3.3.1. While the overall system is generic to any point-based, sparse visual feature, our implementation consists

of SURF visual features (Bay et al., 2008) extracted from both grayscale stereo images and the *Forest CC* color-constant images detailed in Section 3.3.3. Edges in the STPG link vertices with uncertain,

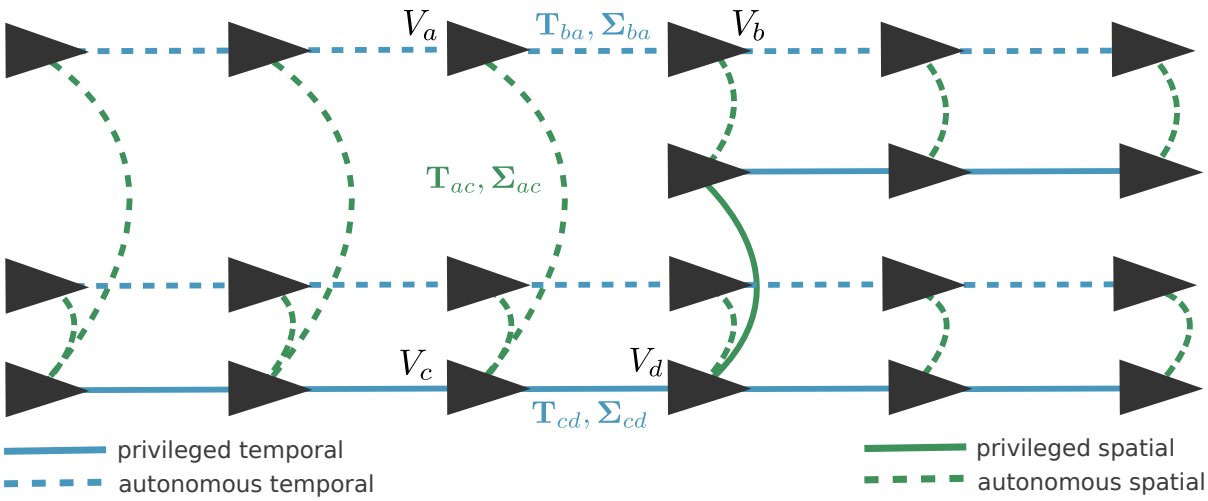


Figure 4.3: Overview of the STPG data structure used to represent our multi-experience network of paths. Experiences are shown as rows of vertices (black triangles) connected metrically through blue temporal edges calculated via VO while the robot is either being manually driven (solid) or autonomously repeating (dashed). Experiences are related metrically through green, spatial edges, calculated through localization and can either be added autonomously while driving (dashed) or manually while adding a branch or loop closure (solid).

relative $SE(3)$ transformations. Temporal edges (blue lines) connect temporally adjacent vertices, while spatial edges (green lines) connect vertices that are spatially close but temporally distant. A set of vertices connected by temporal edges can be generated using a stereo VO pipeline, while spatial edges between vertices require metric localization. Edges are considered *privileged* (solid lines) if the robot was being manually driven, or *autonomous* (dashed lines) if the robot was autonomously repeating a route. Our system uses the STPG to represent a multi-experience network of connected paths, where each *experience* is a collection of vertices linked by temporal edges. The subgraph containing *all* privileged experiences represents the collection of safe, drivable paths. Autonomous experiences linked to this privileged subgraph are used to aid the navigation algorithms by providing a wealth of place-specific information.

4.3.2 Stereo VO Pipeline

The stereo VO pipeline is the core estimation machinery of our system; It creates experiences in the STPG, provides updates of the robot's position at the rate of the sensor, and refines landmark positions and edges in the graph structure. The stereo VO system consists of two parallel estimation pipelines. This configuration, popularized by Klein and Murray (2007) as Parallel Tracking and Mapping (PTAM), allows for uninterrupted, high-rate motion estimation while concurrently optimizing poses and landmarks in the map. This section provides details on the two parallel pipelines deployed in the stereo VO estimator: i) the high-rate, frame-to-keyframe motion estimation pipeline illustrated in Figure 4.4, and ii) the low-rate sliding-window bundle adjustment pipeline illustrated in Figure 4.5.

Frame-Keyframe Visual Odometry

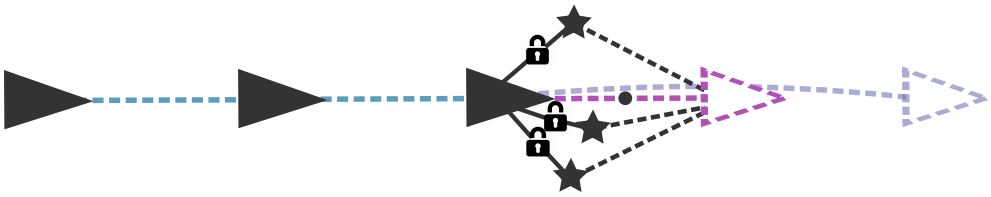


Figure 4.4: The high-rate, frame-to-keyframe VO pipeline. Features are matched from the live frame (light-purple triangle) to the previous keyframe (black triangle). Landmarks (3D feature positions shown as black stars) are locked, and the trajectory (dark-purple line) is optimized to minimize reprojection error in the live frame. Once the trajectory is estimated it can be used to extrapolate the robot's position (light-purple line) from the previous keyframe.

The high-rate, frame-to-keyframe pipeline, shown in Figure 4.4, provides an estimate of the rover's motion between the latest keyframe in the pose graph, V_{kf} (black triangle with stars), and the latest input frame, V_f (purple, dashed triangle), at the rate of the camera. In this state estimation problem, we are optimizing the trajectory (dark-purple, dashed line with dot) between the two frames given measurements in V_f to the locked landmarks originating in V_{kf} (black stars) using the Simultaneous Trajectory Estimation and Mapping (STEAM) engine (Anderson and Barfoot, 2015). The trajectory consists of the $SE(3)$ transformation and uncertainty, $\{\mathbf{T}_{fkf}, \mathbf{\Sigma}_{fkf}\}$, the body-centric velocities $\{V_{kf}, V_f\}$, and the vertex timestamps, $\{t_{kf}, t_f\}$. The initial velocity, $\boldsymbol{\varpi}_{kf}$, is assume to be known and locked. Once the trajectory is estimated, it can be used to extrapolate the rover's position from the previous keyframe to timestamps in the future (light-purple, dashed triangle).

The structure of the frame-to-keyframe pipeline is similar to the pipeline presented in the previous iterations of the VT&R 1.0 autonomous path-following system (Section 2.2). As the MEL system incorporates the multi-channel ideas presented in the MCL system (Chapter 3), the input to the frame-to-keyframe pipeline is a collection of stereo images from all configured information channels.

Extraction/Triangulation The first step of the VO pipeline is to extract and triangulate measurements from each left/right rectified stereo pair. Stereo measurements in the system consist of left-right visual feature keypoints, $\mathbf{Y} = \{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y_n}\}$, with descriptors, $\mathbf{D} = \{\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d_n}\}$, and triangulated 3D positions, $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p_n}\}$. The stereo keypoint $\mathbf{y}_i$ takes the following form:

$$\mathbf{y}_i = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} + \delta\mathbf{y}_i,\ \delta\mathbf{y}_i = \mathcal{N}(\mathbf{0}, \mathbf{R}_i), \tag{4.1}$$

where $\{u_l, v_l\}$ is the left keypoint measurement, $\{u_r, v_r\}$ is the right keypoint measurement, and $\mathbf{R}_i$ is the 4×4 covariance matrix representing the uncertainty of the left-right measurements. This measurement is used to triangulate a 3D position, $\mathbf{p}_i$, with a 3×3 covariance matrix, $\boldsymbol{\phi}_i$, using the inverse stereo camera model and its Jacobian.

Landmark Matching The goal of the landmark matching module is to find observations (dashed, black lines) in the live frame, V_f of the established, locked landmarks (black stars) in the keyframe,

V_{kf}, given the stereo measurements of keyframe landmarks (black lines) extracted from the live frame, $\{\mathbf{Y}_f, \mathbf{P}_f, \mathbf{D}_f\}$, and the trajectory of the previous frame-to-keyframe estimate. To reduce the image search space, an estimated motion, $\{\check{\mathbf{T}}_{fkf}, \check{\mathbf{\Sigma}}_{fkf}\}$, is queried from the previous trajectory and used to project landmarks into the new frame. These reprojected landmarks are then iteratively matched to stereo measurements extracted from the live frame that fall within a local search space via their appearance and geometry. Feature matches between the the live frame and the keyframe are sent through a locally optimized RANSAC (Chum et al., 2003) implementation to remove outliers and provide an initial estimate of the posterior transform between V_f and V_{kf}.

State Estimation Given a set of inlier matches between V_f and V_{kf}, We now seek the optimal posterior motion between V_f and V_{kf}:

$$\{\hat{\mathbf{T}}_{f,kf}, \hat{\mathbf{\Sigma}}_{f,kf}, \hat{\boldsymbol{\varpi}}_f\}, \tag{4.2}$$

given a set of landmarks in $\underrightarrow{\mathcal{F}}_{kf}$, with associated measurements in $\underrightarrow{\mathcal{F}}_{f}$, and a prior term on motion. This is achieved by minimizing the following objective function:

$$J(\mathbf{T}_{f,kf}) = \frac{1}{2}\sum_{j=1}^{M} \mathbf{e}_j{}^T \mathbf{R}_j^{-1} \mathbf{e}_j + \frac{1}{2}\mathbf{e}^T \mathbf{Q}^{-1}\mathbf{e}. \tag{4.3}$$

The first term in J sums the squared error of keyframe landmarks reprojected into $\underrightarrow{\mathcal{F}}_{f}$. Given a landmark, j, with mean, $\mathbf{p}_{kf,j}$, expressed in $\underrightarrow{\mathcal{F}}_{kf}$, and a stereo measurement, $\mathbf{y}_j$, of j expressed in $\mathbf{p}_{kf,j}$, the error term, $\mathbf{e}_j$ is defined as:

$$\mathbf{e}_j = \mathbf{y}_j - \mathbf{g}(\mathbf{T}_{f,kf}\mathbf{p}_{kf,j}), \tag{4.4}$$

where $\mathbf{g}(\cdot)$ is the stereo measurement model. The error function is weighted by the inverse of the covariance matrix, $\mathbf{R}_j$, which describes the keypoint measurement noise.

The second term of J smooths the continuous-time trajectory between V_{kf} and V_f given the following Gaussian Process (GP) prior distribution on possible trajectories:

$$\dot{\mathbf{T}}(t) = \boldsymbol{\varpi}^{\wedge}\mathbf{T}(t), \tag{4.5}$$

$$\dot{\boldsymbol{\varpi}} = \mathbf{w}(t), \qquad \mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c\delta(t-t')), \tag{4.6}$$

where $\{\mathbf{T}(t), \boldsymbol{\varpi}(t)\}$ is the pose and velocity of the robot at measurement time, t, respectively, and $\mathbf{w}(t)$ is a stationary, zero-mean, white-noise GP with power-spectral density matrix, $\mathbf{Q}_c$, and $\delta(\cdot)$ is the Dirac delta function. Where the operator, $\wedge$, transforms $\boldsymbol{\varpi} \in \mathbb{R}^6$ into a 4×4 member of the *Lie Algebra*, $\mathfrak{se}3$ as detailed in (2.4) and the operator, $\vee$, is its inverse. The error term, $\mathbf{e}$ is then defined as:

$$\mathbf{e} = \begin{bmatrix} \ln(\mathbf{T}_{f,kf})^{\vee} - (t_f - t_{kf})\boldsymbol{\varpi}_{kf} \\ \boldsymbol{\mathcal{J}}(\ln(\mathbf{T}_{f,kf})^{\vee})^{-1}\boldsymbol{\varpi}_f - \boldsymbol{\varpi}_{kf} \end{bmatrix}, \tag{4.7}$$

where $\ln(\mathbf{T}_{f,kf})^{\vee}$ converts the $SE(3)$ transformation to its Lie algebra vector, and $\boldsymbol{\mathcal{J}}(\cdot)$ is the left Jacobian of $SE(3)$. The inverse covariance matrix, $\mathbf{Q}^{-1}$, weights the error function appropriately for the chosen $\mathbf{Q}_c$. To obtain an optimal posterior estimate, $\hat{\mathbf{T}}_{bd}$, (4.14) is iteratively linearized and refined

in a nonlinear least-squares optimization using our STEAM engine. For more details on our trajectory estimation see Anderson and Barfoot (2015). If the translational or rotational motion of the posterior estimate is large, or the number of matched features between V_f and V_{kf} drops too low, V_f is inserted as a new vertex in the graph; otherwise, it is discarded.

Keyframe Bundle Adjustment

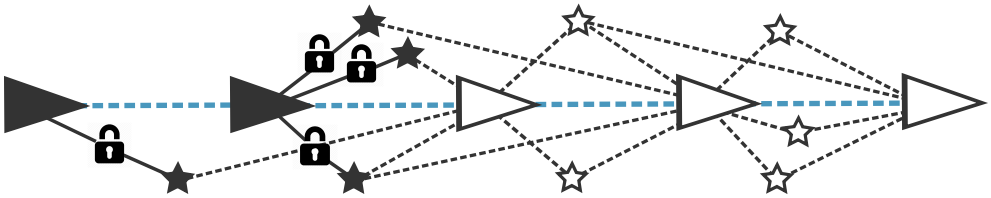


Figure 4.5: The low-rate, sliding-window keyframe bundle adjustment pipeline. Landmark positions and transforms in the window (white stars and blue lines) are optimized by minimizing the reprojection error in each observing keyframe and deviation from the prior.

Following insertion of a vertex, a graph optimization problem is performed on a sliding window containing the latest vertices in the live experience (Figure 4.5), again using STEAM (Anderson and Barfoot, 2015). In this bundle adjustment problem, we are optimizing the window of poses and landmarks (white triangles and stars) as well as the continuous-time trajectory starting at the beginning of the first pose whose landmarks are observed in the window (first black triangle). Poses and landmarks outside of the window (black triangles and stars) are locked. All poses in the optimization problem are first relaxed into a single coordinate frame with the origin being the earliest vertex whose landmarks are observed in the bundle adjustment window (leftmost black triangle). Landmark positions remain relative to the vertices in which they originate. This problem uses the same state estimation machinery detailed in the frame-to-keyframe method to iteratively refine the estimation of the trajectory and landmark positions. Upon successful optimization, the poses, landmarks, and their uncertainties are updated in the graph.

4.3.3 Multi-Experience Localization (MEL)

The MEL algorithm, illustrated in Figure 4.6, provides metric localization with respect to a privileged path (solid blue line) across seasonal changes through the use of the multi-experience STPG data structure. The MEL algorithm estimates the posterior transform and uncertainty, $\{\hat{\mathbf{T}}_{bd}, \hat{\mathbf{\Sigma}}_{bd}\}$ (purple, dashed line), between the most recent vertex in the live experience, V_b, and the estimated closest vertex in the privileged experience, V_d. This process is initiated upon creation of a new vertex in the live experience, and occurs directly after keyframe bundle adjustment during autonomous traversals. This process minimizes the reprojection error of landmarks in a window of recommended experiences (green rectangles)[2] that are observed by V_b. Throughout the algorithm we make use of the prior term, $\{\check{\mathbf{T}}_{bd}, \check{\mathbf{\Sigma}}_{bd}\}$, obtained by compounding the uncertain transforms (Barfoot and Furgale, 2014),

$$\{\mathbf{T}_{ba}, \mathbf{\Sigma}_{ba}\},\ \{\mathbf{T}_{ac}, \mathbf{\Sigma}_{ac}\},\ \{\mathbf{T}_{cd}, \mathbf{\Sigma}_{cd}\}, \tag{4.8}$$

[2]Because the MEL algorithm is computationally intractable as the number of experiences grows, an algorithm to select the best N experiences must be used to keep the algorithm bounded. While the topic of choosing which experiences to use is not a novel contribution of this thesis, it is being researched in parallel with this thesis and is presented at a high level in Chapter 5, which provides details on the multi-experience autonomous path-following system, VT&R 2.0.

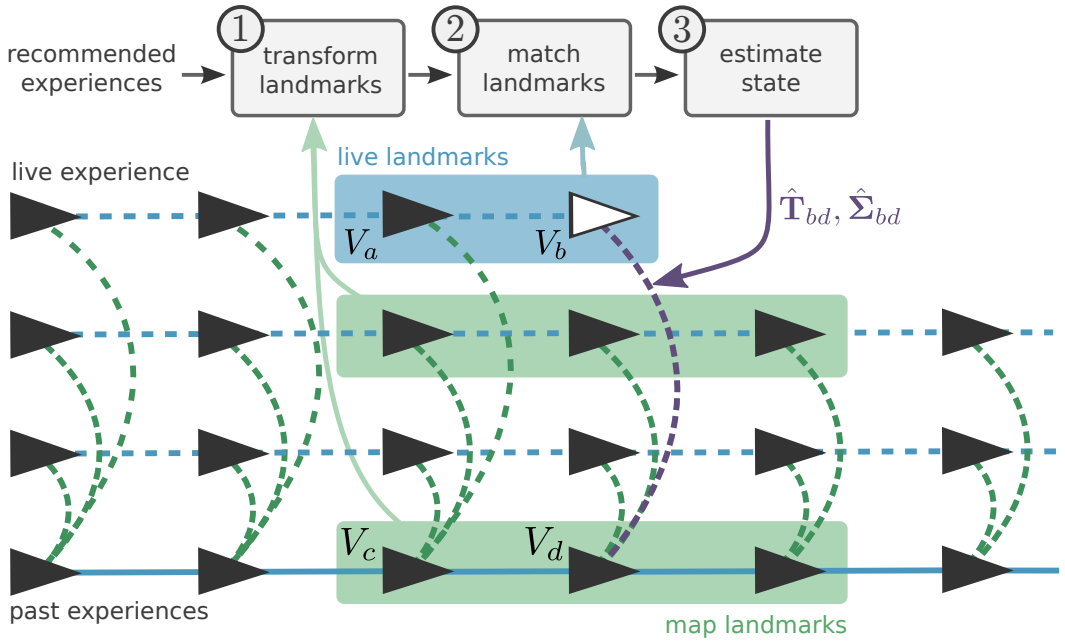


Figure 4.6: An overview of the MEL algorithm. Given a selection of experiences to localize against (green rectangles), the algorithm solves for the transformation between the vertex in the live experience, V_b, and the vertex in the map experience, V_d. The algorithm begins by transforming all landmarks originating from vertices within the localization windows of selected experiences (green rectangles) to the coordinate frame of V_d. Next, landmarks in the live vertex, V_b, are matched to these transformed map landmarks in a breadth-first-search pattern starting at the target vertex, V_d. Finally, the transformation $\mathbf{T}_{bd}$ (purple, dashed line) is estimated by performing a simple keyframe-to-keyframe bundle adjustment problem with map landmark positions locked.

which are computed through previous VO and localization estimates. The remainder of this section provides details on the components of the MEL pipeline: a) Landmark Transformation, b) Multi-Experience Matching, and c) Multi-Experience State Estimation.

Landmark Transformation

The first step of MEL is to transform all landmark means and uncertainties originating from vertices within the map window of recommended experiences from their respective coordinate frames to $\underrightarrow{\mathcal{F}}_d$, the coordinate frame of V_d and the one in which localization is to be computed. Given a 3D landmark expressed in some vertex map frame, $\underrightarrow{\mathcal{F}}_m$, with mean and covariance, $\{\mathbf{p}_m, \mathbf{\Phi}_m\}$, the transformation to $\underrightarrow{\mathcal{F}}_d$ is given by:

$$\mathbf{p}_d = \mathbf{T}_{dm}\mathbf{p}_m \tag{4.9}$$

$$\mathbf{\Phi}_d = \mathbf{D}^T \mathbf{p}_d^{\odot} \mathbf{\Sigma}_{dm} \mathbf{p}_d^{\odot T} \mathbf{D} + \mathbf{D}^T \mathbf{T}_{dm} \mathbf{D} \mathbf{\Phi}_m \mathbf{D}^T \mathbf{T}_{dm}^T \mathbf{D}, \tag{4.10}$$

where $\odot$ is an homogeneous-point operator (Barfoot and Furgale, 2014) given by

$$\begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix}^{\odot} = \begin{bmatrix} \eta \mathbf{1} & -\boldsymbol{\epsilon}^{\wedge} \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix}, \tag{4.11}$$

with

$$\boldsymbol{\epsilon}^{\wedge} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & 0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \tag{4.12}$$

and $\mathbf{1}$ is the identity matrix. This process is carried out on all landmarks in the map window to produce a set of landmarks with 3D position and uncertainty, all expressed in the privileged frame, $\underrightarrow{\mathcal{F}}_d$. While it may seem superfluous to transform the locations and uncertainties of landmarks that are not yet known to be inlier matches, these help refine the matching process, making it faster and more robust.

Bookkeeping the uncertainties is a critical aspect of making MEL work well; the bridging experiences are daisy-chained over time from the privileged experience, so while they may have more matches to the live experience (than those directly from the privileged experience) due to more similar appearance, the spatial uncertainty of those matches may be higher when transformed to the privileged frame, $\underrightarrow{\mathcal{F}}_d$. Keeping track of all the uncertainties ensures we properly weight all the matches in our localization.

Multi-Experience Matching

The goal of multi-experience matching is to associate every landmark in V_b to a landmark in the map window. The process begins with labeling all landmarks in the live vertex as unmatched. Vertices in the map window are sequentially examined starting from V_d in a breadth-first-search pattern. We chose to center the search around the privileged target vertex as a heuristic for prioritizing landmarks that have the lowest uncertainty in the target privileged frame. For every new vertex visited, the transformed map landmarks associated with this vertex are projected into the camera frame of vertex V_b using the prior term, $\{\check{\mathbf{T}}_{bd}, \check{\boldsymbol{\Sigma}}_{bd}\}$. Each landmark associated with this vertex is then checked for matching feasibility to the unmatched live landmarks by comparing keypoint position and descriptor appearance. This process continues until one of three criteria are met: i) a sufficient number of matches are found, ii) the amount of time has surpassed the allowance, or iii) the map window is exhausted. As the process of comparing visual features is costly and the size of the map window grows linearly with experiences, this process is the most computationally expensive step of multi-experience localization. Upon completion of landmark matching, the problem is set up so that there are associated 3D landmarks in the coordinate frames of V_b and V_d. This information is sent through a locally optimized RANSAC (Chum et al., 2003) implementation to remove outliers and provide an initial estimate of the posterior transform between V_b and V_d. This step is identical to the VO outlier rejection process outlined in Section 4.3.2, with the exception of the source of the landmarks.

Multi-Experience State Estimation

Given a set of map landmarks in the coordinate frame of V_d and a set of inlier observations to these landmarks from the live vertex, V_b, we now seek the optimal posterior,

$$\{\hat{\mathbf{T}}_{bd}, \hat{\boldsymbol{\Sigma}}_{bd}\}, \tag{4.13}$$

given the prior term, $\{\check{\mathbf{T}}_{bd}, \check{\mathbf{\Sigma}}_{bd}\}$, as well as associated data between V_b and map landmarks in the coordinate frame of V_d. This can be achieved by minimizing the following objective function:

$$J(\mathbf{T}_{bd}) = \frac{1}{2}\sum_{j=1}^{M} \mathbf{e}_j{}^T \mathbf{R}_j^{-1} \mathbf{e}_j + \frac{1}{2}\mathbf{e}^T \mathbf{R}^{-1} \mathbf{e}. \tag{4.14}$$

The first term in J sums the squared reprojection error of map landmarks. Given a map landmark, j, with mean and uncertainty, $\{\mathbf{p}_{d,j}, \mathbf{\Phi}_{d,j}\}$, expressed in the coordinate frame of V_d and a stereo measurement of j, $\mathbf{y}_j$, with uncertainty, $\mathbf{Y}_j$, expressed in the camera frame of V_b, the reprojection error is defined by

$$\mathbf{e}_j = \mathbf{y}_j - \mathbf{g}(\mathbf{T}_{bd}\mathbf{p}_{d,j}), \tag{4.15}$$

$$\mathbf{R}_j = \mathbf{Y}_j + \mathbf{Z}_j \tag{4.16}$$

$$\mathbf{Z}_j = \mathbf{G}_j \mathbf{T}_{bd} \mathbf{D} \mathbf{\Phi}_{d,j} \mathbf{D}^T \mathbf{T}_{bd}^T \mathbf{G}_j^T, \tag{4.17}$$

where $\mathbf{g}(\cdot)$ is the stereo measurement model and $\mathbf{G}_j$ is its Jacobian (evaluated at $\mathbf{p}_{b,j} = \mathbf{T}_{bd}\mathbf{p}_{d,j}$). This weights each error by uncertainty in the measurement and the map. The second term of (4.14) constrains the optimization problem by the prior with

$$\mathbf{e} = \ln(\check{\mathbf{T}}_{bd}\mathbf{T}_{bd}^{-1})^{\vee}, \quad \mathbf{R} = \check{\mathbf{\Sigma}}_{bd}, \tag{4.18}$$

where $\vee$ is the inverse operator of $\wedge$ Barfoot and Furgale (2014). To obtain an optimal posterior estimate, $\hat{\mathbf{T}}_{bd}$, (4.14) is iteratively linearized and refined in a nonlinear least-squares optimization using our STEAM engine (Anderson and Barfoot, 2015). For a thorough review of information on nonlinear optimization we refer the readers to Chapter 4 of Barfoot (2017). Specifically, we make use of the Dogleg Gauss Newton algorithm (Powell, 1964) for nonlinear optimization with the Dynamic Covariance Scaling (DCS) (Agarwal et al., 2013) robust cost function used on landmark cost terms and the L2 robust cost function used on the prior term. In the absence of any matches between the live image and map, the prior estimate (based on VO) is returned.

4.4 Experimental Setup

This section details the set up of the experiments designed to validate the MEL algorithm. This section presents two offline experiments and a vision-in-the-loop experiment designed to validate different aspects of the localization algorithm. Extended field tests of the MEL algorithm are reserved for the next chapter, where we present a multi-experience vision-in-the-loop autonomous path following system. The first offline experiment was conducted to quantify the impact the autonomous bridging experiences have on localization performance when the appearance of the live experience is significantly different from the privileged. The second offline experiment was designed to compare the MEL algorithm to its most related predecessor, Experience-Based Navigation (EBN) (Churchill and Newman, 2013). The vision-in-the-loop experiment was designed to quantify the spatial drift incurred from localizing indirectly to the privileged experience as the number of bridging experiences increases.

4.4.1 CSA Offline Analysis

To assess the impact of adding bridging experiences in the MEL algorithm, a series of experiments were conducted offline using the stereo-imagery data set collected during field testing of the lighting-resistant VT&R 1.0 system presented in Section 3.4.2. The full set of data consists of a single 1 km teach pass covering a wide variety of environments (simulated Mars terrain, grassy field and wooded paths; see Figure 4.7), which is then followed by 26 autonomous repeats on the same path over the same and following days (see Table 4.1). Appearance changed significantly due to sunny conditions with harsh shadows on the first day and overcast weather on the second, as well as terrain modification from the vehicle (tire tracks). The autonomous repeats were performed using the lighting-resistant MCL system detailed in Section 3.3.3 and the robot maintained path-following autonomy for over over 99.9% of the route. For these experiments, however, we configure the MEL system to use only the grayscale stereo channel to challenge the performance of the framework over a subset of the repeat runs.

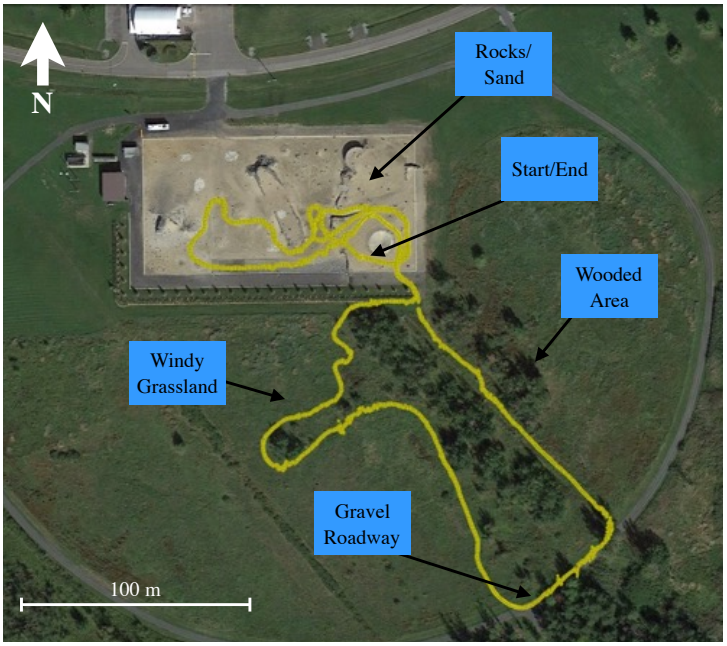


Figure 4.7: Satellite imagery of the Canadian Space Agency's Mars Emulation Terrain and surrounding woodland. A 1km route was driven 27 times across a wide variety of lighting conditions to gather the data set used for evaluation.

The offline experiments consisted of simulating localization using varying numbers of experiences from the CSA data set, (Table 4.1). The first experience, e0, is the manually driven, privileged experience, while the remainder are autonomously driven experiences (e1-e6, e16, e27) of the same route. Details of the localization experiments (the sets of experiences) are listed in Table 4.2.

The first set of experiments (g0-g5) analyze the performance of localization between the privileged experience (e0) and an experience gathered approximately seven hours later (e6) with an increasing number of bridging experiences. We choose to focus on e6 for the reason that it has the worst localization performance against e0 (even when tested with colour-constant imagery in previous experiments) due to significant lighting changes. Making the assumption that the change is roughly equal throughout the day, we add experiences in a mean-splitting pattern as a simple heuristic; e.g., e6 is localized against e0

Table 4.1: Overview of the experiences in the CSA data set.

ID	Start Time	Duration [hh:mm]	Δt [hh:mm]	Sky Condition
e0	2014/05/12 10:35	00:34	00:00	sunny
e1	2014/05/12 11:40	00:28	01:05	sunny
e2	2014/05/12 12:53	00:27	02:18	sunny
e3	2014/05/12 13:35	00:26	03:00	sunny
e4	2014/05/12 14:55	00:31	04:20	sunny
e5	2014/05/12 16:06	00:32	05:31	cloudy
e6	2014/05/12 17:27	00:26	06:52	sunny
e16	2014/05/13 11:00	00:27	24:25	cloudy
e27	2014/05/15 08:50	00:25	70:31	sunny

Table 4.2: Overview of the graph configurations used for multi-experience localization evaluation.

ID	Live experience	Privileged experience	Bridge experiences
g0	e6	e0	–
g1	e6	e0	e3
g2	e6	e0	e2, e4
g3	e6	e0	e1, e3, e5
g4	e6	e0	e1, e2, e4, e5
g5	e6	e0	e1, e2, e3, e4, e5
g6	e27	e0	–
g7	e27	e0	e1, e2, e4, e5
g8	e27	e0	e1, e2, e4, e5, e16

in test g0, then e3 is added as the bridging experience, etc. This is continued until test g5, where all bridging experiences are used to localize e6. Identifying the optimal set of bridging experiences is not a goal of this thesis; we instead aim to show how such experiences can be used effectively.

The next set of experiments (g6-g8) test the performance of e27 as the live experience. The experience e27 is the most temporally distant to e0 (by more than 70 hours) and contains significant terrain modification due to the robot carving deep troughs in the forest environment and other robots creating tire tracks in the sand in the MET. To test localization to e27, we introduce three experiments using an increasing number of bridging experiences. In experiment g6, we test using no bridging experiences. In experiment g7, we add a set of experiences that capture the lighting change seen in the first day, (e1, e2, e4, e5). Finally, in experiment g8, we use the aforementioned set as well as one experience from the second day, e16, that captures overcast conditions and terrain modification. We hypothesize that adding an extra experience with overcast conditions will significantly increase performance with the rationale that overcast conditions are easy to localize against, regardless of lighting conditions. Results of this offline experiment can be found in Section 4.6.1.

4.4.2 Offline EBN Comparison

The MEL localization system is heavily inspired by the EBN localization system developed by Churchill and Newman (2013). The primary difference between the two localizers is that the MEL system uses *all* available experiences to solve one localization problem, while the EBN system uses all available experiences to solve *many* localization problems, choosing the top performer as the state estimate. This many-to-one state estimation scheme is a core contribution of the MEL algorithm. In collaboration with the Oxford Robotics Institute (ORI), the publishers of the EBN system at the University of Oxford, we designed an experiment to fairly compare the performance of both algorithms in an offline setting.

To show the benefits of the MEL algorithm, we directly compare the two systems using the data set collected during the UTIAS multi-season field test. This data set and vision-in-the-loop field test that created it is presented fully in Section 5.3.4. It consists of repeated autonomous traversals of a 165 m loop in a meadow at UTIAS and spans four months. Examples of appearance change in this data set can be seen in Figure 4.8. In this experiment, both the MEL and EBN systems build multi-experience maps and provide metric localization on the first 75 loops of this data set, which occurred between the dates of 01/31/17 and 02/19/17, where the appearance of the scene dramatically varied due to snow fall, freezing rain, and snow melt.

Figure 4.8: Examples of appearance change primarily due to winter weather observed over four months while autonomously following a path at the University of Toronto during the field test described in Section 5.3.4.

To isolate the primary differences between both systems, the localization configurations are as similar as possible. In particular, both system's make use of upright SURF visual features from grayscale and color-constant images, with the notion of localization failure kept the same (inlier match count >= 6 inliers). To eliminate the impact of experience selection on the performance of both localizers during this experiment, *all* past experiences are used during localization. While this setup is not feasible for real-time, vision-in-the-loop operations, it is suitable for this offline comparison. Through this experiment, we show that the MEL algorithm's ability to perform many-to-one localization is a benefit when the appearance of the scene significantly diverges from what is already in the map. Detailed results of this comparison can be found in Section 4.6.2.

4.4.3 Photocopy-of-a-Photocopy

When localizing with the MEL algorithm, as the appearance of the live view diverges from the privileged experience, *all* of the matched landmarks will be to autonomous experiences. These landmarks are transformed to the privileged frame using uncertain localization transforms with some amount of error. Spatial drift will occur when there are errors in the localization estimates used to transform the landmarks. This field test was designed to quantify the effects of this spatial drift. To do this, we

Figure 4.9: Setup for the photocopy-of-a-photocopy experiment. This experiment was conducted to help quantify the spatial drift incurred in MEL algorithm. Translational ground truth was collected with a Leica Totalstation tracking a prism on a Clearpath Grizzly RUV.

conducted a simple field test, depicted in Figure 4.9. In this field test, the robot was manually demonstrated a straight, 50 m path and autonomously traversed the path path back and forth 180 times while collecting ground truth on its position with a Leica Total Station. To simulate the worst-case scenario for spatial drift growth, we forced the system to localize to only the most recent experience for every traverse. At any point in time during the autonomous traversals, this forces the MEL algorithm to use map landmarks with the worst localization errors. To quantify the effects of spatial drift over time, we analyze the ground-truth error between the manually taught path and the autonomous traversals. Results of this field test can be found in Section 4.6.3.

4.5 Evaluation Metrics

To evaluate MEL using the aforementioned experiments, we selected three metrics: a) Cross-track uncertainty, b) Feature inlier count, c) Computation time, and d) Root Mean Squared Error (RMSE).

4.5.1 Cross-track uncertainty

This is our primary metric for judging localization success. We define cross-track uncertainty as the one-standard-deviation uncertainty of our lateral translation estimate relative to the privileged path. This tells us how uncertain we are to the left or the right while following the privileged path, and can be directly interfaced with our path-tracking controller to provide safe autonomous driving based on

the lateral constraints of the path. It is important to note that while uncertainty is calculated at every stage of the algorithm from keypoint detection to landmark transformation, we have not yet performed a rigorous evaluation of our uncertainty estimates with respect to ground truth to ensure consistency. Therefore we treat this metric as a way to compare relative performance between experiments and do not necessarily trust the exact scale of our uncertainty estimates.

4.5.2 Feature inlier count

This metric is simply the number of inliers observed (after RANSAC) at each localization point over the entire traverse for each experiment. In the offline experiment, this metric evaluated as a Cumulative Distribution Function (CDF) provides a measure of how much each experience adds to the state estimation problem by examining experiments g0-g5. In the online experiment, evaluating the median inlier count for each repeat highlights the ability to localize in real-time even with the addition of more experiences. For localization success, we require at least 10 self-consistent feature matches (i.e., after RANSAC) or fall back to the prior alone (i.e., VO).

4.5.3 Computation time

As complexity of the MEL algorithm scales linearly with the number of experiences in the worst case scenario (when matching reaches the upper bound of time), we are interested in observing the average computation time of localization for each experiment. In order for this algorithm to support vision-in-the-loop route following, this solve time needs to be fast enough to support the incoming localization requests from the path tracker. We currently have this set to 250ms, which is based on the average rate of vertex additions to the graph, primarily based on distance driven by the robot.

4.5.4 Root Mean Squared Error (RMSE)

To quantify the amount of spatial drift observed in the Photocopy of a Photocopy (PoaP) experiments, we look at the Root Mean Squared Error (RMSE) in translation between the ground truth of the privileged teach traversal and all subsequent autonomous repeat traversals.

4.6 Results

4.6.1 CSA Offline Analysis

We begin our analysis with results on the offline multi-experience localization experiments described in Section 4.4.1. The goal of this section is to quantify the impact of using bridging experiences in the MEL algorithm when localizing a live experience to a privileged experience seven hours apart in time where the appearance of the scene is significantly different due to lighting.

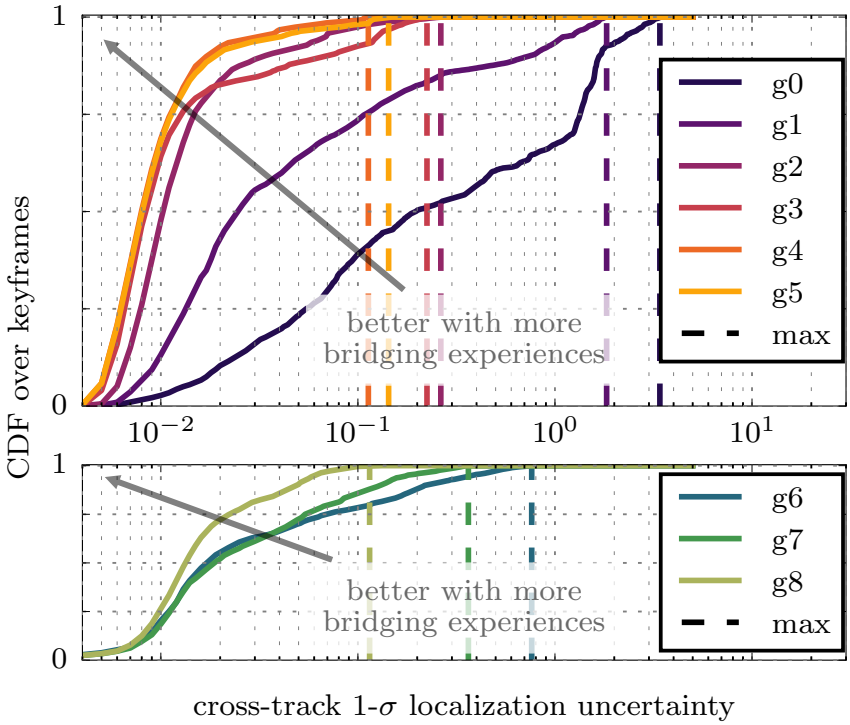


Figure 4.10: Cumulative distribution (over keyframes) of the cross-track (left-right) localization uncertainty. This figure reads: *for Y fraction of the traverse, the robot's* 1σ *cross-track uncertainty was less than X meters.* The uncertainty units are in meters, though it is a relative measure (see Cross-track uncertainty). The first set of experiments, g0–g5 (see Table 4.2), show dramatic improvement with the addition of the first two bridging experiences, and very little improvement beyond. The second set of experiments, g6–g8, also shows significant improvement with more bridging experiences.

Cross-track uncertainty

Results of all offline experiments with respect to cross-track uncertainty are presented in Figure 4.10. This shows the CDF of the cross-track uncertainty for the entire traverse for each experiment. In the worst case scenario (g0, single-experience localization), the robot would have driven with a maximum cross-track uncertainty of approximately 3.5 m. Only 70% of the traverse would be driven with a cross-track uncertainty less than 1 m. This is in stark contrast with experiment g2, which uses bridging experiences spaced evenly two hours apart from each other. Maximum cross-track uncertainty observed in g2 never exceeded 0.3 m. Additionally, 90% of the traverse was driven with a cross-track uncertainty of less than 0.05 m. This is shown in Figure 4.12a with example images of some sections highlighted in Figure 4.12.

The results from g0-g5 indicate that when appearance is gradually changed between the live and privileged view, adding more bridging experiences improves the localization performance. The dramatic increase in performance between g1 and g2 indicates that, in sunny conditions, an experience every two hours is most likely sufficient for autonomous navigation. Results from g6-g8 further show this trend. It is interesting to note that the single-experience experiment, g6, has a generally lower uncertainty than the single-experience experiment, g0. This is because despite e27 being the farthest in time from the privileged experience, e0, it is the closest in appearance; e27 took place at 08:50, three days later, when it was also sunny outside and only about two hours earlier (but on a different day). This result confirms the intuition behind the Time of Day (ToD) experience selector (Section 5.2.5) which attempts to prioritize experiences that are closest in time of day rather than absolute time. Results from g8 show that the addition of an overcast experience greatly increases performance.

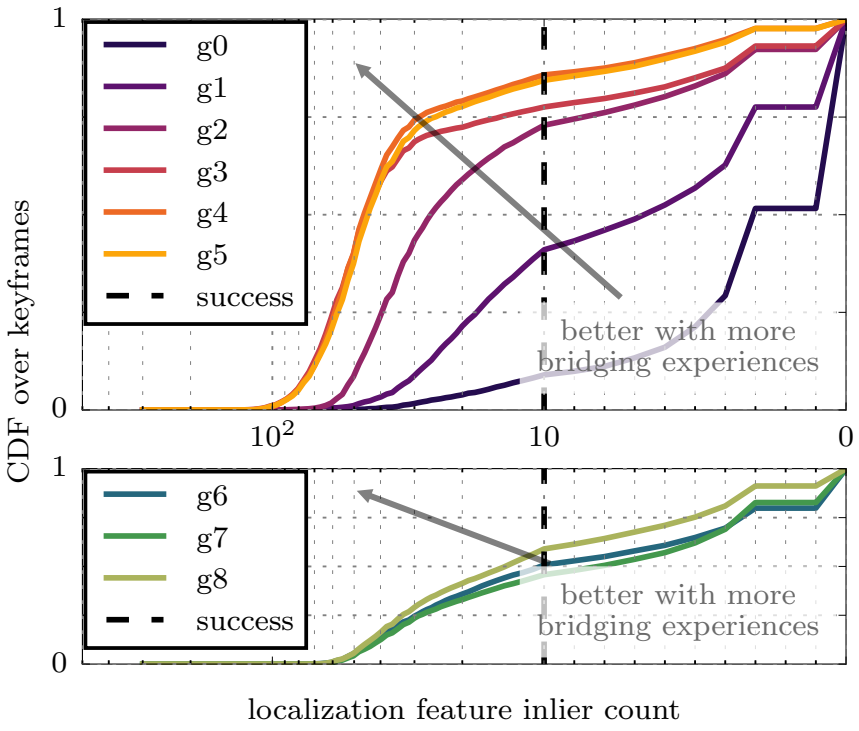
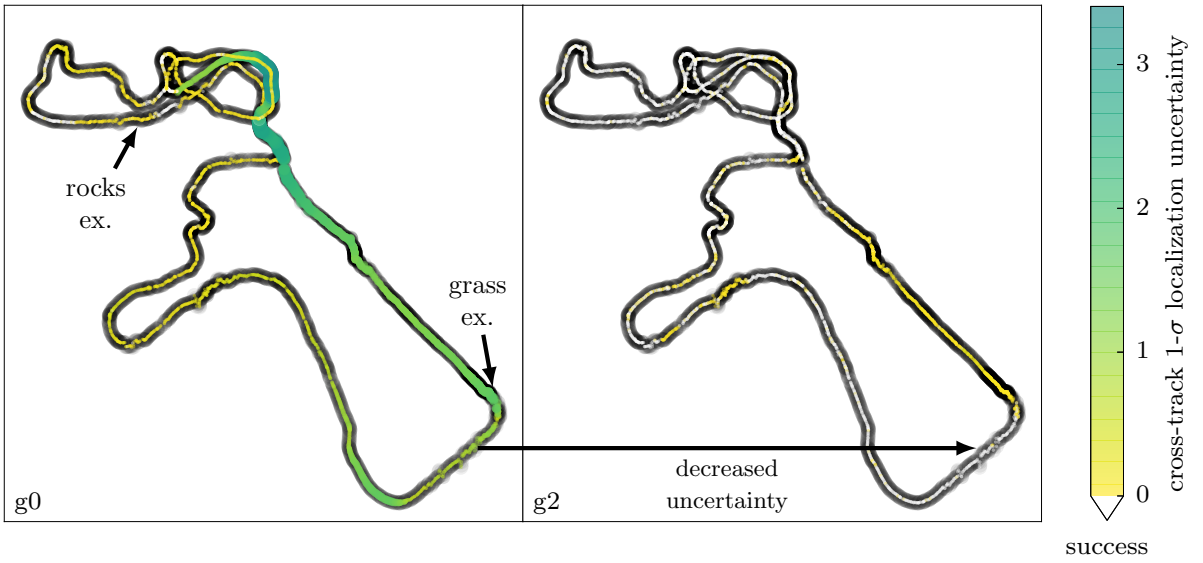


Figure 4.11: Cumulative distribution (over keyframes) of the number of inlier matches to any previous experience being used. We require at least 10 inliers (dashed line) to accept the localization as a success, which is why g0 performed so poorly in the uncertainty metric; only 10% of the time did it have enough inliers to be accepted. By far the most dramatic improvement is adding one (g1) or two (g2) bridging experiences when trying to localize e6.

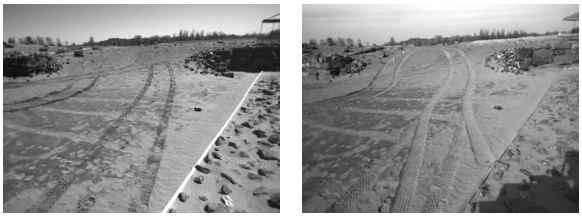
Feature inlier count

The feature inlier counts with respect to the offline experiments are presented in Figure 4.11. This figure shows the CDF of inlier matches found between the live view and experience map for each experiment. The number of inliers found is highly correlated with the cross-track uncertainty. Therefore, the order of inlier-match performance across experiments is similar to that seen in the cross-track uncertainty results.



(a) Cross-track 1σ uncertainty (coloured) for two interesting experiments (g0 and g2) overlaid on the GPS path of the vehicle (black, with a 3 m margin), with successful localizations coloured white. The top section of the path is in a rocks-and-sand environment, while the bottom section is in vegetation. The section of the path that remains difficult even for g2 is in 1 m tall grass (see Figure 4.12b), with tall trees on either side that cast long shadows. Images from the rocks and grass examples can be see in Figure 4.12.

(b) The most difficult section of the experiment (Tall Grass).



(c) An easy section of the experiment (rock piles).

Figure 4.12: Example images each showing e0 (left) and e6 (right), showing (a) grass (hard) and (b) rocks (easy) localization situations.

Computation Time

Timing analysis is presented in Figure 4.13. This figure shows the interquartile range of the computation time (in milliseconds) of localization for each experiment. The trends in the figure reflect the early-stopping criterion of the multi-experience matching algorithm. When a sufficient number of matches to the map is found, the algorithm exits early. This correlates to the boxes that represent Q1-Q3 of the interquartile range, which range from 25-75 ms. When there are not enough matches found, either the graph is exhausted, or the time allowance runs out. In the case of these experiments, this matching value is set to 125 ms (half of the 250 ms for the full MEL update). This can be seen in the whiskers, which range from 100-150 ms and represent the 99th percentile.

It is interesting to note that despite having a smaller number of bridging experiences, g1 and g2 have higher solve times than g3-g5. This is due to the fact that g3-g5 have consistently higher inlier counts, which triggers early stopping in the matching stage more often. The increase in computation time between g6 and g7 can be explained by the addition of extra experiences that do not aid localization. Computation time between g7 and g8 are similar because the number of inliers seen in both cases

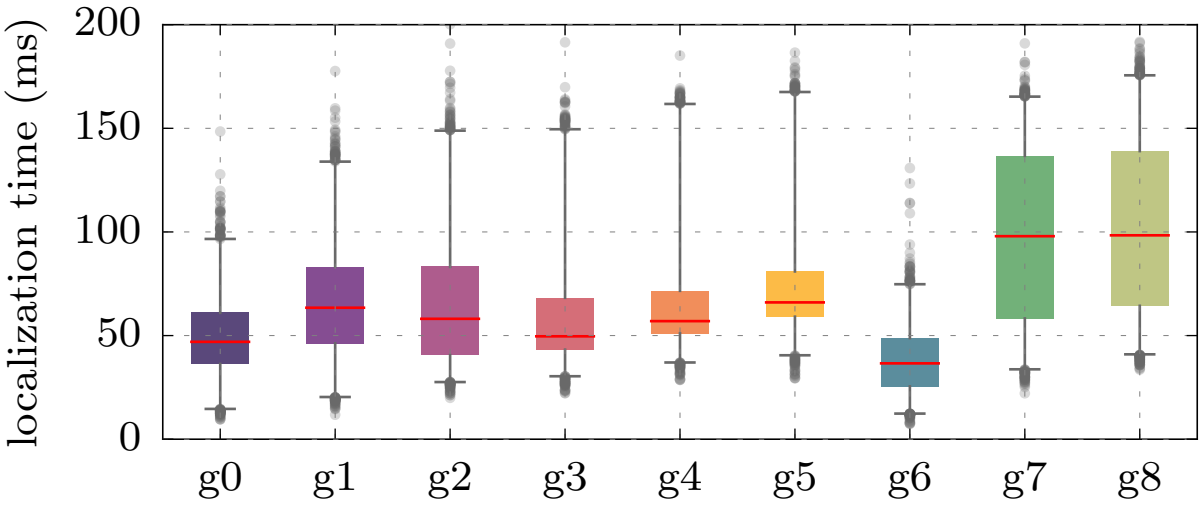


Figure 4.13: Interquartile range of localization computation time for each experiment. There is a noticeable increase from using no intermediate experience (g0, g6), to adding one or two The algorithm has linear complexity with the number of experiences used, so eventually computational

is relatively low, which indicates the early-stopping match criterion was not triggered. While these timing results show that a graph containing several experiences can be run in real time, the complexity of the algorithm will eventually be a limiting factor to how many experiences can be processed. To computationally bound the MEL algorithm, a method to select the best N experiences for a localization solve can be used. The topic of choosing which experiences to use is *not* a novel contribution of this thesis, rather it is research being conducted in parallel to this work (MacTavish and Barfoot, 2014), and presented in Chapter 5 at a high level as a component of the long-term autonomous path-following system, Visual Teach & Repeat (VT&R) 2.0.

Conclusions

This subsection presented the results of the performance of the MEL algorithm with respect to the offline data set experiments detailed in Section 4.4.1. The primary objective of these experiments was to demonstrate the MEL algorithm's ability to perform metric localization to a privileged experience using several intermediate bridging experiences gathered during autonomous operation. The results validated this claim, showing the system's ability to provide adequate localization between a privileged view and a live view captured seven hours apart in time on a sunny day, where the appearance of the scene drastically changed due to lighting. These results furthermore provide a guide for online use of this algorithm in an autonomous path-following system, showing that when more than two experiences are used, separated roughly two hours apart in time, the inlier match count remains above 60 for the entire traverse with the $1-\sigma$ cross track uncertainty below 0.3 m.

These experiments furthermore show that the computation time of the algorithm can remain within the threshold of real-time operation when a small number of experiences are used. These results were originally published in Paton et al. (2016), which presented the MEL algorithm.

4.6.2 Offline EBN Comparison

This section provides results on the offline experiment to compare the performance of the MEL localizer to the primary influence of this thesis' work, the EBN localization system (Churchill and Newman, 2013). In this comparison, both algorithms build multi-experience maps and perform metric localization on the first 75 autonomous traversals of the UTIAS multi-season field test detailed in Section 5.3.4, covering the first 12 km of autonomous driving. To perform a fair comparison, both system's use SURF visual features in their respective state estimation pipelines and recommend *all* previous experiences in localization. We compare the performance of the two localization systems by analyzing the distance the robot would have driven on dead reckoning.

Distance on dead reckoning

Results with respect to distance on dead reckoning are presented in Figure 4.14 and Figure 4.16. Figure 4.14 shows the maximum distance driven on dead reckoning for all test traverses for each algorithm. This figure shows that the maximum distances driven on dead reckoning for the MEL algorithm remained below 2 m for all but three autonomous traverses, where the maximum values rose to 5, 7, and 8 m. This is in contrast to the performance of the EBN algorithm, which incurred higher values during

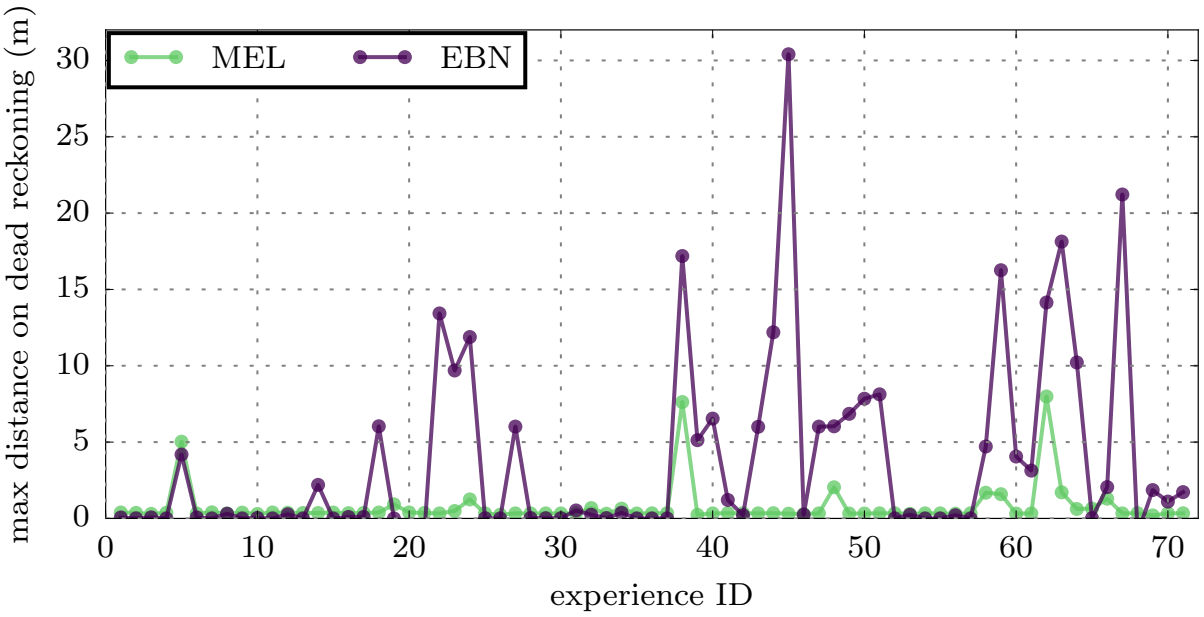


Figure 4.14: Maximum distance traveled on dead reckoning for both algorithms on the first 75 runs of the UTIAS multi-season data set.

three sections of the experiment: i) experiences 20-30, ii) experiences 38-51, and iii) experiences 58-68. During these periods, the maximum distance driven on dead reckoning exceeded 12, 30, and 21 m. These sections of the experiment are notable for containing a large amount of appearance change. Examples of this rapid appearance change is shown in Figure 4.15. Between experiences 20 and 30 (top row), light snow transitioned to freezing rain, and then melted to expose the dead vegetation of the meadow. Between experiences 38 and 51 (middle row), the meadow was covered in a blanket of deep snowfall. Between experiences 58 and 78 (bottom row), all of the snow in the meadow melted and sky conditions

were sunny to reveal dead vegetation and shadows. During these times of difficult appearance change, the MEL algorithm greatly benefits from being able to find data correspondences in all previous experiences to perform its localization estimate.

Figure 4.15: Rapid appearance change examples in the UTIAS multi-season field test. Each row shows the rapid appearance change over the course of a few days. *top row:* experiences 20-30, the appearance of the scene changes from light snow fall to freezing rain to snow melt. *middle row:* experiences 38-51, the appearance of the scene changes from little snow to light snow to heavy snow in four days. *bottom row:* experiences 58-68, the appearance of the scene changes from heavy snow to snow melt in four days.

Figure 4.16 shows the CDF of the distance driven on dead reckoning for every autonomous traverses compared between the two algorithms. The EBN results (top figure) show that for all but 10 traverses, the robot drove less than 0.1 m on dead reckoning for at least 90% of each traverse. For the other 10 traverses, this value lies between 55% and 85%. These correspond to the traverses in Figure 4.14 with the highest maximum values, occurring during extreme appearance change. The MEL results show that for all traverses, the robot drove less than 0.1 m on dead reckoning for at least 86% of each traverse, and drove less than 1 m for 95% of each traverse.

Conclusions

This section provided a quantitative comparison of the performance of two algorithms, Multi-Experience Localization (MEL) and Experience-Based Navigation (EBN) (Churchill and Newman, 2013), with respect to vision-based, metric localization across extreme seasonal appearance change. In this experiment we compare both algorithms' abilities to localize the first 75 autonomous traverses of the 165 m loop of the UTIAS multi-season data set. These traverses contain appearance change due to heavy snowfall, freezing rain, night-time driving, and snow melt.

In this comparison, we demonstrate that both algorithms provide impressive localization considering the difficulty of the data set. For the majority of autonomous traverses, both algorithms provide

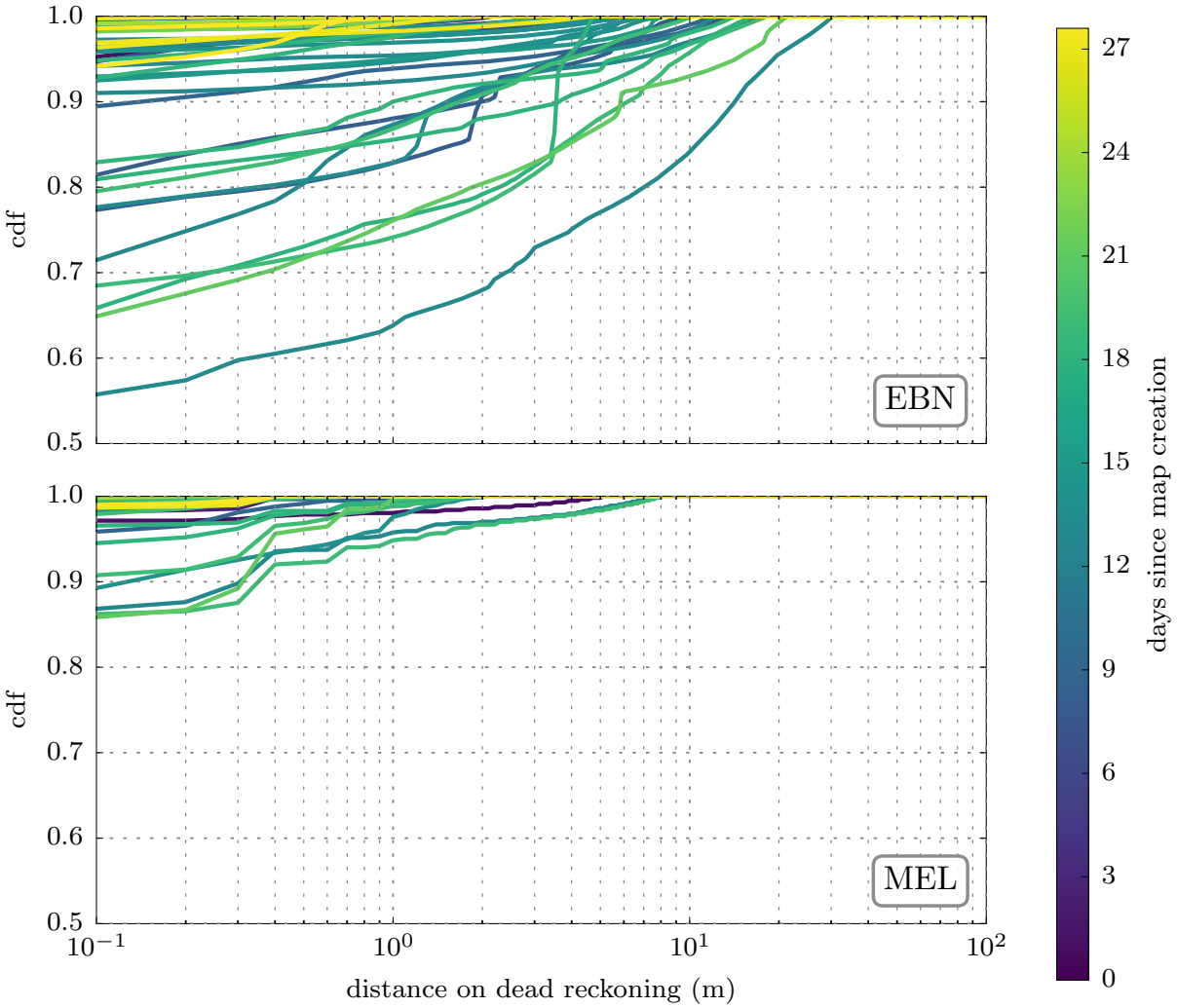


Figure 4.16: CDF of the distance traveled on dead reckoning for both algorithms on the first 75 runs of the UTIAS multi-season data set.

adequate localization for path tracking requirements. However, for traverses where there is rapid appearance change from what the robot has previously experienced, the MEL algorithm provides more stable localization. This is primarily due to the MEL algorithm's ability to use data correspondences from *all* previously gathered bridging experiences in a single state estimate. In this many-to-one localization scheme, a handful of feature matches from each experience can provide sufficient inliers where it would otherwise fail.

4.6.3 Photocopy of a Photocopy

The previous experiment demonstrated that the MEL algorithm is capable of metric localization across significant appearance change using multiple autonomous bridging experiences. This section presents results on the PoaP field test detailed in Section 4.4.3. This test was designed to exacerbate the effects of spatial drift in the MEL localization system while observing the robot's ground truth position with an external sensor. Results of the field test show that spatial drift in the MEL system is very modest as

the number of bridging experiences grows.

Ground Truth RMSE

Ground truth results of the field test are presented in Figure 4.17. This figure shows the Root Mean Squared (RMS) translational error between the privileged, manually driven pass and each autonomous traverse. For the first 40 traverses, the RMSE hovered near 5 cm and then gradually rose to 8-9 cm during the next 40 traverses. The error of the first 40 traverses is likely the baseline path-tracking error, and the rise between experiences 40 and 80 is likely when the spatial drift began to show. After this initial climb, the error slowly ascended to between 9-10 cm for the remainder of the field test.

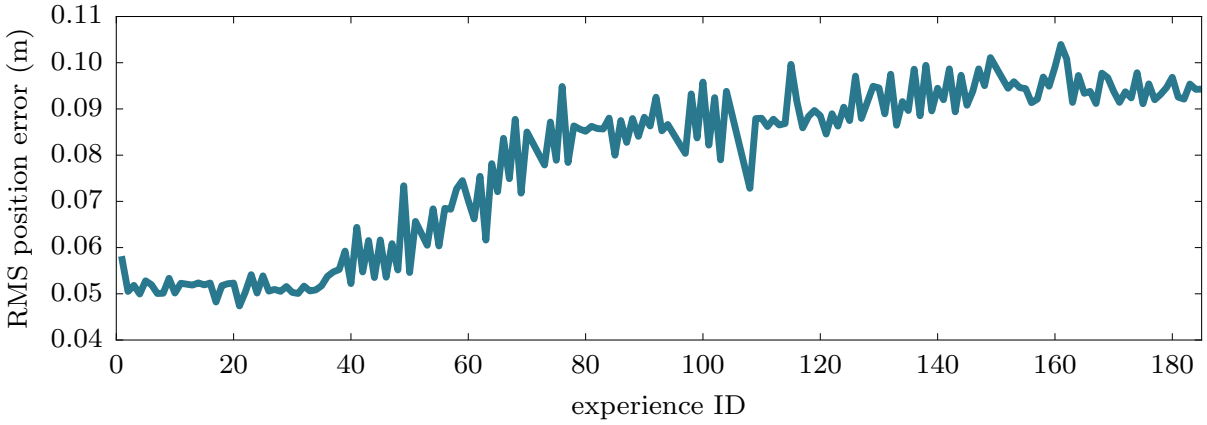


Figure 4.17: RMSE position error between the teach pass and all autonomous traverses in the photocopy-of-a-photocopy field test.

In this field test, spatial drift is exacerbated by forcing the MEL algorithm to only consider the most recent bridging experience in the localization estimate. This spatial drift is observable in the ground truth errors; however, even in this worst-case scenario the positional error while autonomously traversing this path is within the bounds of safe driving for most applications.

Localization Results

During the field test, the robot continuously traversed the small path back and forth. Because of the extremely small temporal disparity between the live view and the most recent bridging experience in the map, the localization performance was excellent throughout the field test. This can be seen in the inlier feature match results presented in Figure 4.18. The results show that for the entire field test, the median inlier match count remained near 200 inliers, with the min value near 150 matches.

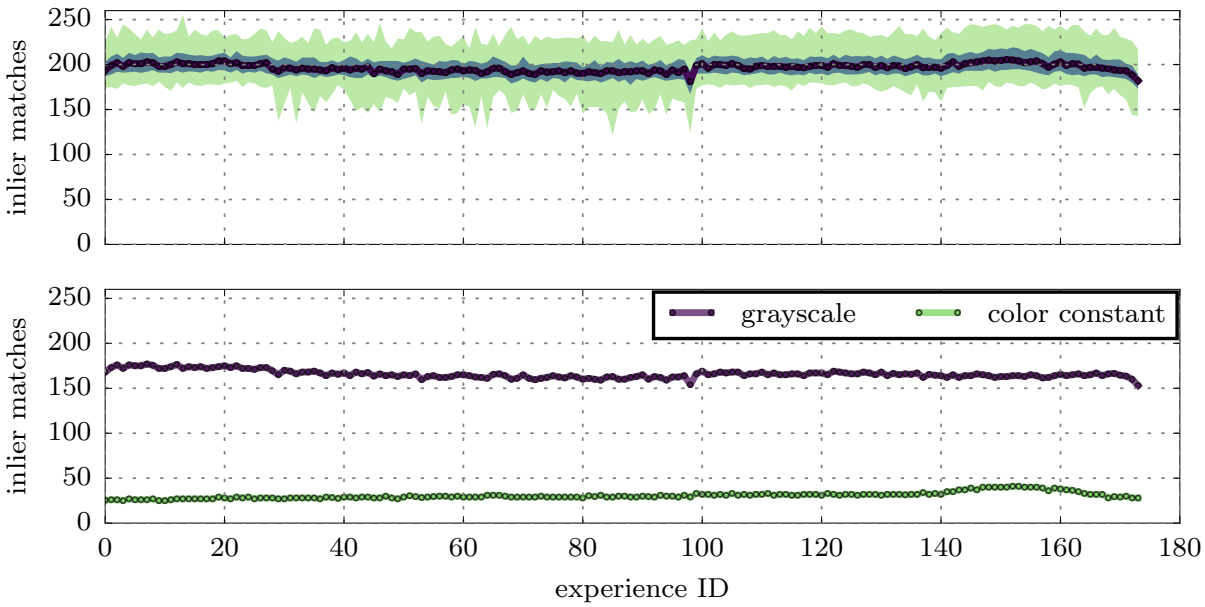


Figure 4.18: Inlier Match count for the PoaP Field Test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel. These results show that for all autonomous traverses in this field test, the inlier match count remained nominal, with a median value of 200 inliers from all channels.

Conclusions

The objective of this field test was to better understand the impact of spatial drift on the performance of autonomous path following in the MEL system. In a simple experiment, the robot autonomously repeated a small 50 m straight path back and forth 185 times in a row. While doing so, the spatial drift in the system was exacerbated by forcing the localizer to only consider the most recent bridging experience in the state estimate. To observe spatial drift in this worst-case scenario, we recorded the ground truth position of the robot with a Leica TotalStation. The results of this field test demonstrate the MEL algorithm's ability to use bridging experiences to localize with respect to the privileged experience and reliably follow the manually taught path. In the nominal configuration of the MEL algorithm—when many experiences are used to localize, we expect the spatial drift to be even lower than what was observed during this experiment.

4.7 Summary and Novel Contributions

This chapter presented the Multi-Experience Localization (MEL) algorithm that provides metric localization across extreme appearance change between a live experience and a privileged experience through bridging experiences gathered during autonomous traversals. The MEL localizer was first published in the proceedings of the international conference on Intelligent Robots and Systems (IROS) (Paton et al., 2016). In summary, this chapter presents the following novel contributions:

1. A data structure that relates multiple experiences together metrically.
2. A methodology to metrically localize a live experience to a privileged, manually driven experience using several intermediate experiences gathered during autonomous operation.
3. A methodology to bookkeep uncertainties in the multi-experience localizer, accounting for uncertain map landmarks originating from multiple experiences.
4. Experimental evaluations of the MEL system to validate the core ideas of metric localization using many experiences.

The results presented in this chapter validate the core ideas behind the MEL algorithm. In Section 4.6.1, we quantified the impact of using bridging experiences and demonstrated confident localization using the MEL algorithm. In Section 4.6.2 we compared the MEL algorithm to its most related work: Experience-Based Navigation (EBN) (Churchill and Newman, 2013), and demonstrated the benefits of the MEL algorithm's many-to-one multi-experience localization scheme when the appearance of the scene rapidly changes. Finally, in Section 4.6.3, we quantified the effects of spatial drift on path following in the MEL system as the number of bridging experiences in the map increases. However, in order for this algorithm to be computationally tractable, it needs to be used in conjunction with an experience selection algorithm. While experience selection is not a topic of research for this thesis, it is used in our full VT&R 2.0 autonomous path-following system, presented in the next chapter. With the VT&R 2.0 system, we will show that if the MEL algorithm is given a sufficient number of experiences, it is possible to localize across appearance change as drastic as green spring grass vs deep snow, enabling long-term, vision-based path following.

Chapter 5

Multi-Experience VT&R

In the previous chapter we presented MEL—a long-term, vision-based localization algorithm designed specifically for autonomous path-following systems. In this chapter, we integrate MEL into the autonomous path-following system, Visual Teach & Repeat (VT&R) 2.0, and demonstrate the system's ability to perform long-term, vision-in-the-loop path following through extensive field testing, covering over 178 km of vision-in-the-loop autonomous driving. An example of VT&R 2.0 in the field is shown in Figure 5.1.

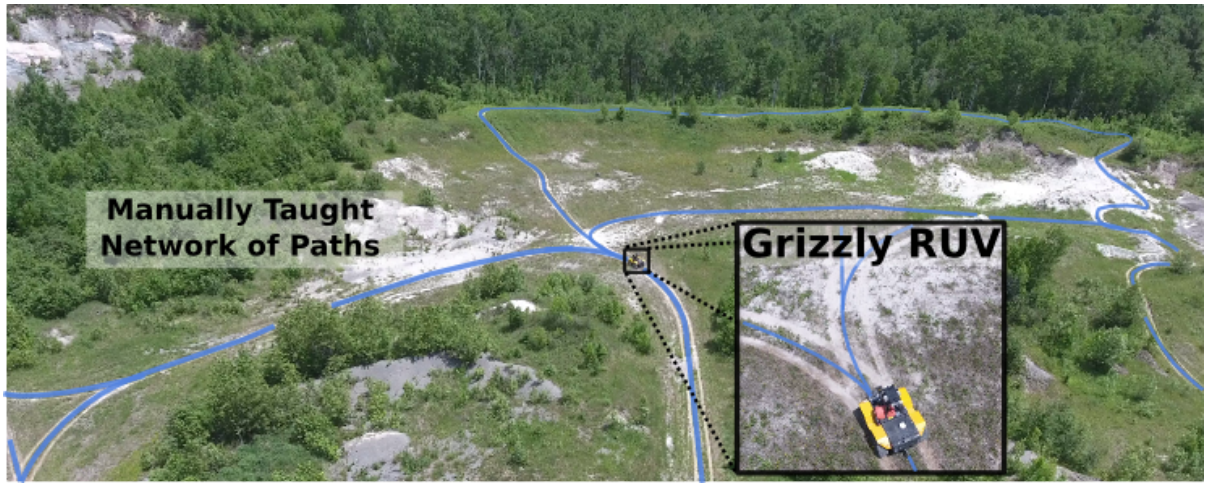


Figure 5.1: A Grizzly RUV deployed with the VT&R 2.0 autonomous path-following algorithm navigating a 5 km network of manually taught paths. Using the MEL algorithm presented in the previous chapter, the robot autonomously traversed the network continuously for eleven days, accumulating over 140 km of vision-in-the-loop driving with an autonomy rate of 99.59% of distance traveled while experiencing significant appearance changes in the environment. Through extensive field testing, this chapter demonstrates the VT&R 2.0 system's ability to perform large-scale, long-term autonomous path following using passive vision sensors.

5.1 Introduction

Autonomous path-following algorithms allow robots to repeat networks of connected paths previously driven by human operators using only on-board sensors. The unique task of autonomously traversing

a human-taught path gives the robot a strong prior on where it is safe to drive (Berczi and Barfoot, 2016). This allows for confident, autonomous navigation through rough, outdoor terrain that would otherwise be inaccessible or require complex, generic, and potentially risky terrain-assessment algorithms. Furthermore, these methods can be implemented to have bounded computation costs and minimal map sizes (Furgale and Barfoot, 2010), making them well suited for long-range navigation. These benefits make autonomous path-following appealing for industrial applications that consist of repeated traversals over constrained paths, such as factory floors, orchards, and mines. They are also well-suited to applications that consist of autonomous exploration and retrotraverse such as search-and-rescue and hazardous-exploration robots. However, autonomous path-following systems suited for these applications need the ability to navigate large-scale environments over long time periods. Furthermore, they require constant, metric localization to the manually driven path as the input error signal to a path-tracking controller to ensure minimal drift, and the ability to recognize and cope with obstacles blocking the path. These requirements pose a serious challenge for vision-based systems whose advantages of cost and commercial ubiquity come at the expense of robustness to appearance change.

In this chapter, the Multi-Experience Localization (MEL) algorithm is integrated into the vision-in-the-loop autonomous path-following system, Visual Teach & Repeat (VT&R) 2.0, and experimentally validated through a series of field tests. These field tests cover over 178 km of vision-in-the-loop autonomous driving and demonstrate the system's ability to: i) autonomously traverse a large-scale network of paths over a period of two weeks experiencing appearance change due to lighting and weather, ii) autonomously traverse a path taught in the daytime continuously over an entire diurnal cycle, including at night with on-board headlights, and iii) autonomously traverse a path over a period of 4 months experiencing seasonal appearance change such as snow fall, rain, snow melt, and vegetation growth. All field tests were performed with autonomy rates of over 99% and demonstrate our system's ability to perform reliable, long-term navigation using only a single stereo camera. The novel contributions in this chapter are: i) a vision-in-the-loop autonomous path-following system that makes use of a multi-experience localization and mapping framework to provide inter-seasonal autonomy and nighttime autonomy with on-board headlights, and ii) extensive long-term field tests of the system involving autonomy in unstructured, outdoor environments with rapidly changing appearance, covering over 178 km of vision-in-the loop autonomous driving. The contributions of this chapter have been accepted and are to appear in the proceedings of the international conference on Field and Service Robotics (FSR) (Paton et al., 2017a) in September, 2017.

5.2 Methodology

This section presents details of the multi-experience autonomous path-following system, VT&R 2.0, capable of autonomous navigation on large-scale networks of connected paths across long time periods. The system is based on MEL, the novel multi-experience localization and mapping framework that provides metric localization with respect to a manually driven experience across extreme appearance change (Chapter 4). The section begins with a high level overview of VT&R 2.0 and then provides details on the following components of the system: i) network construction, ii) route planning, iii) autonomous path following, iv) experience selection techniques.

5.2.1 System Overview

The VT&R 2.0 system enables vehicles to autonomously navigate large-scale networks of connected paths over long time periods. At a high level, the system consists of a teaching mode where a human operator constructs or adds to a network of connected paths by manually driving the robot and a repeating phase where the robot is autonomously traversing to a goal or a collection of goals in the network. A network of paths in the system is represented by the Spatio-Temporal Pose Graph (STPG), a multi-experience, topometric pose-graph detailed in Section 4.3.1. An experience in this data structure can be thought of as the appearance of the scene the robot experiences while traversing a path and is represented as the output of the system's stereo VO estimator (presented in Section 4.3.2). An experience in the map is labeled as either *privileged* if it was added while teaching a path, or *autonomous* if it was added while repeating a path. Privileged experiences in the network are created while the system is in teach mode and connected to each other with relative $SE(3)$ transformations. These connections are generated while *branching* from an existing path or *merging* into an existing path, forming a loop closure. This subgraph of connected, privileged paths form the core network of paths demonstrated to the robot and is used during autonomous traversals to plan routes through the network. Details on network construction and route planning can be found in Section 5.2.2 and Section 5.2.3, respectively.

To autonomously traverse a route provided by the planner, the robot continuously localizes its current position to a small section of the privileged network map that it is closest to. This is performed by iterleaving stereo VO and metric localization using the Multi-Experience Localization (MEL) algorithm, the primary contribution of this thesis. The MEL algorithm provides long-term, metric localization between the live view and the privileged experiences in the network through the use of bridging experiences in the map that are gathered during autonomous operation. This algorithm furthermore adapts the multi-channel localization framework presented in Chapter 3 to make use of both grayscale and color-constant stereo images in the localizer. At the frame rate of the sensor, the robot's position with respect to the path is sent to a path-tracking controller to follow the path. Details on autonomous path following are presented in Section 5.2.4. To keep the MEL algorithm computationally tractable as the STPG increases in size, an experience selection algorithm curates the map and recommends a small subset of experiences most likely to aid localization. Experience selection in the VT&R 2.0 system is not a contribution of this paper and is only briefly mentioned to gain a full understanding of the system's working. A brief summary of the experience selection methods used in the VT&R 2.0 system is provided in Section 5.2.5. The VT&R 2.0 system overview concludes with a discussion on the parallelization of the state-estimation pipeline in Section 5.2.6.

5.2.2 Network Construction

The VT&R 2.0 system enables the construction and autonomous traversal of networks of connected paths. A network of paths is constructed by computing and connecting privileged experiences to each other using the stereo VO pipeline while the system is in the human-operated teach mode. This process is executed through the user interface shown in Figure 5.2, which provides an intuitive means to construct networks of paths, and command autonomous traversal to goals on the networks. A network is built by adding a teach goal (left panel) and manually driving the robot; this adds privileged experiences to the STPG. If the network exists prior to the teach, then a multi-experience localization search centered around the robot's topological state estimate is performed using the MEL algorithm. Upon successful

localization, the $SE(3)$ transformation between the live view and the closest vertex in the privileged experience is obtained and a privileged spatial edge connecting the two is created, *branching* the new experience off the existing network. The robot will then add vertices and temporal edges to this new experience through the stereo VO pipeline (Section 4.3.2) as it drives. Live experiences can be *merged* back into the network through loop closures initiated through the UI. Given the live experience has been demonstrated and the robot has driven close to an existing area of the network, the operator selects the region of the network closest to the robot in the UI, and the system attempts to localize the live view to the privileged vertices in that region. Upon successful localization, the candidate position of the robot is presented to the operator to confirm. This localization is constantly updated, allowing the user to make minor adjustments before they are satisfied with the loop closure. Confirmation will add a privileged spatial edge from the live vertex to the target. Otherwise, the user may continue driving to improve alignment, or cancel the merge. While loop closures in this relative, topometric data structure will look disconnected if the network is displayed in a single coordinate frame, a relaxed subsection of the network across a loop closure will remain smooth (Van Es and Barfoot, 2015), allowing for confident path tracking. An illustration of the branching and merging process is shown in Figure 5.3.

5.2.3 Route Planning

An autonomous traversal begins by planning a route in the UI by adding a repeat goal (active in Figure 5.2) to the queue, and selecting a sequence of waypoints for the robot to traverse. To plan the path, the system uses the safe, privileged subgraph of the network, including privileged experiences, and privileged spatial edges from branching and merging. Given the robot's current topological position in the network and a set of waypoints, the planner finds the minimum-cost path that covers all selected waypoints in sequential order. Two edge costs we find useful are the path distance, and the temporal age between the live experience and the experience closest in time (combining absolute time and time-of-day). Since our system localizes against multiple autonomous experiences, planning over recently traversed edges is a heuristic to improve the likelihood of successful localization.

5.2.4 Path Following

Given a planned route through the privileged network, path following begins by creating a new autonomous experience, and attempting to localize the first vertex of the new experience to a vertex in the privileged path. This process is identical to the first step of branching, except the added spatial edge is flagged as autonomous. Once the new autonomous experience is connected to the privileged experience using the MEL algorithm, the system begins the process of continuously estimating the vehicle's position relative to the planned route using the stereo VO pipeline to propagate its position relative to the path forward. At the frame rate of the sensor, this information is sent to a model-predictive path-tracking controller (Ostafew et al., 2016) which sends drive commands to the robot to follow the path. When a new vertex is added to the live experience from the VO pipeline, it is localized to the closest vertex in the privileged path using the MEL algorithm (Chapter 4). Upon successful localization, an autonomous spatial edge is added between the two vertices. If localization fails, then the robot's position is propagated forward using VO.

Before being sent to the robot, commanded velocities computed by the path tracker are first inspected by a safety monitor. In addition to limiting commanded velocities to safe speeds, the safety monitor also

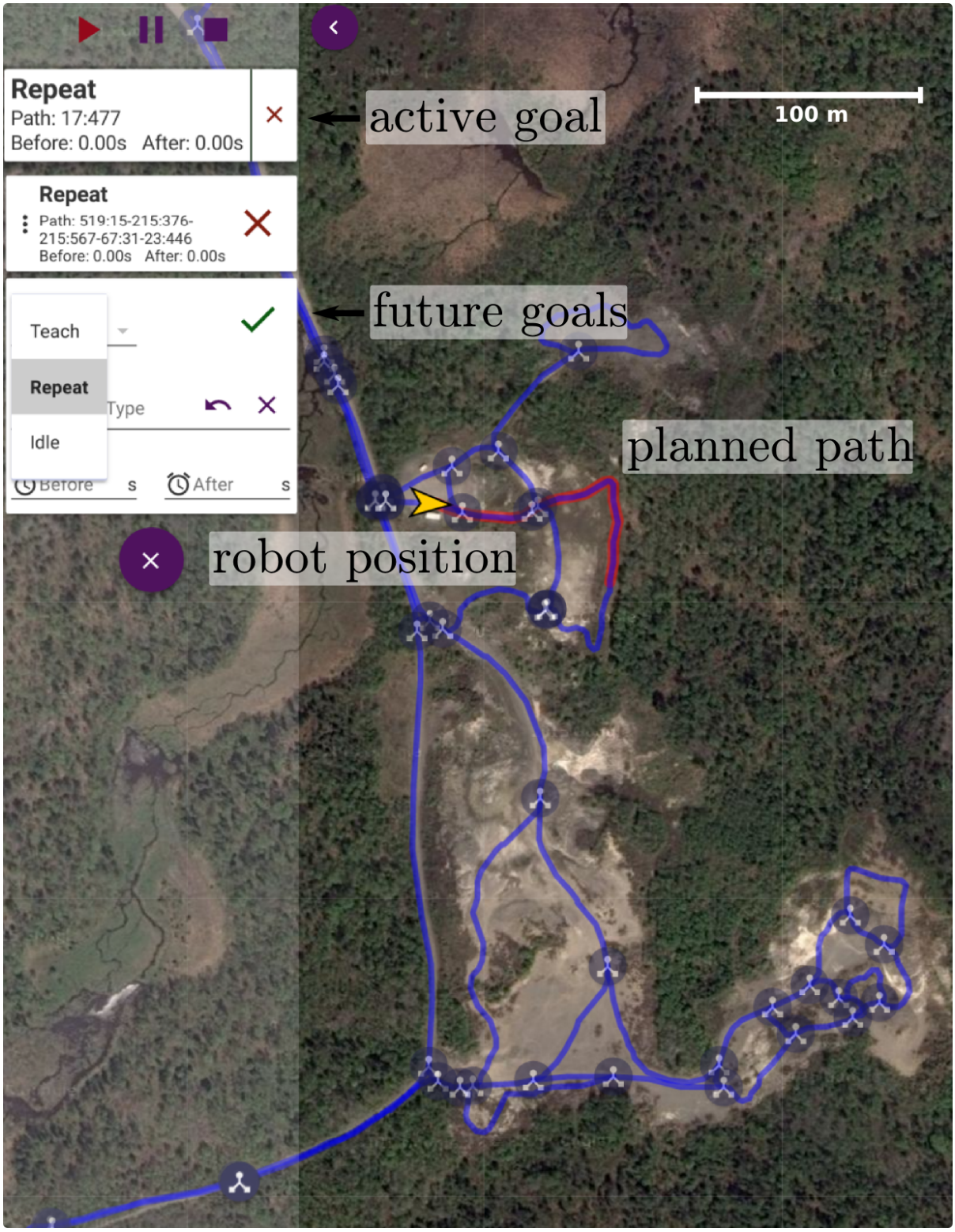


Figure 5.2: Overview of the VT&R 2.0 user interface used to build networks of connected paths and command the rover to autonomously traverse the network.

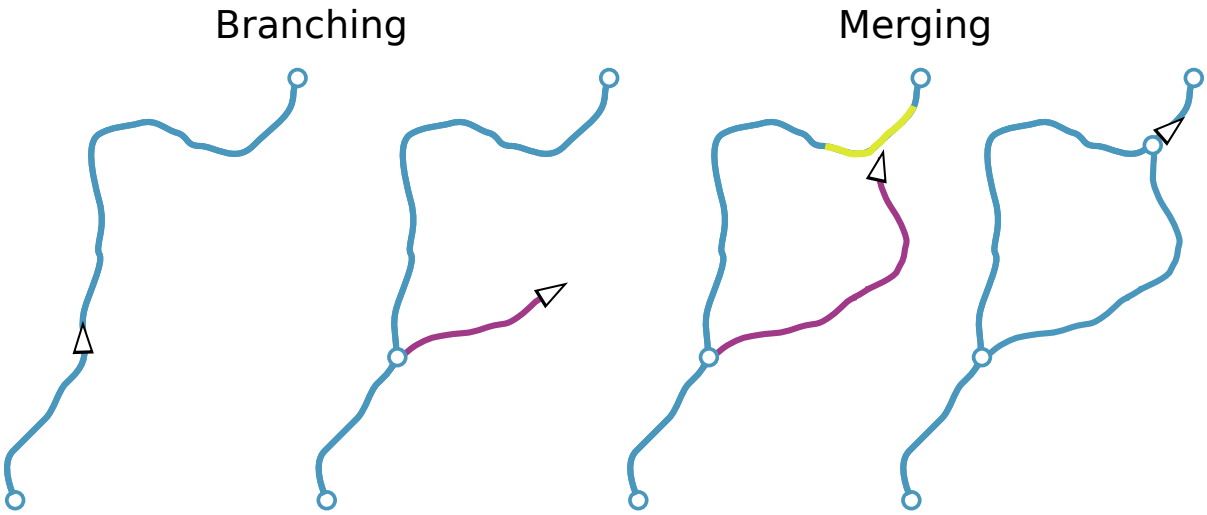


Figure 5.3: Example of the human-operated network construction process in the VT&R 2.0 system. Construction consits of two phases: *Branching* and *Merging*. This example demonstrates the use of branching and merging to add a path to a pre-existing network. *left:* The robot (black triangle) autonomously traverses to a user-specified location in the network. *center left:* The human operator triggers the VT&R 2.0 system to initiate branching mode, and begins driving the robot. This creates a new branch in the network (purple line) by adding a new privileged experience in the STPG through the stereo VO pipeline. *center right:* In order to form a loop closure in the network, the human operator triggers the VT&R 2.0 system to initiate merging, by selecting an area of the network they are close to (yellow line) This triggers localization between the robot view and the designated area of the network. *right:* Once localization is successful, the human operator triggers the VT&R 2.0 system to complete the merge, which forms a loop closure, and establishes the new branch as a part of the network.

considers deadman controls, localization performance, and place-dependent terrain assessment (Berczi and Barfoot, 2016) to ensure safe driving. Localization safety checks are based on the uncertainty of the vehicle's estimate with respect to the privileged path. If the uncertainty grows larger than the user-specified threshold for a given dimension, the safety monitor halts the robot and transitions the VT&R 2.0 system into a relocalization process. This will start a localization search starting at the last known position in the network. If relocalization is successful, then the robot will continue autonomously traversing the path. If relocalization fails, then the vehicle may attempt to halt the traverse, save the current VO output as a new experience in the graph, and use it to autonomously traverse back to a safe location in the network.

5.2.5 Experience Selection

This section provides information related to the experience selection algorithms used in VT&R 2.0 to computationally bound the MEL algorithm by limiting the amount of experiences used in localization. The objective of these experience selection algorithms is to find the subset of experiences that are most likely to contribute landmark matches in the MEL algorithm. There are *no novel contributions* related to this thesis presented in Section 5.2.5—rather these methods are contributions of work being performed in conjunction with this work. We present the high level information of these algorithms in this thesis for completeness of the system description as they are in use during all of the VT&R 2.0 field tests.

Time of Day Experience Selection

Over moderate time periods such as weeks and months, and disregarding weather effects, the appearance of the scene of a given day will most likely look the most similar to its appearance at the same time of day in past days. This is because over modest time scales, the elevation and azimuth of the sun will remain similar, meaning the placement of shadows and lighting will also remain similar. This simple intuition is the core idea of the time-of-day experience selector.

Given the set of total experiences, the time-of-day experiences selector finds the top N scoring experiences according to the following criterion:

$$\mathbf{score}(e_i) = \alpha \mathbf{td}(e_l, e_i) + \mathbf{tod}(e_l, e_i), \tag{5.1}$$

$$\alpha = \frac{1.0}{7.0 * 24.0}, \tag{5.2}$$

where e_i is the ith experience, e_l is the live experience, $\mathbf{td}(\cdot)$ computes the total time elapsed between the two experiences in seconds, and $\mathbf{tod}(\cdot)$ computes the time-of-day difference for between the two experiences. The scalar α down weights old experiences whose appearance will have changed due to weather and seasonal effects. This simple and computationally efficient experience selector is suitable for short-term operation where there is no weather or seasonal change or when the robot is in near-constant operation, guaranteeing that the appearance of the scene between the live experience and the most recent experiences will be gradual. This method does not work when temporal loop closures are required to continue localization. This occurs when the appearance of the scene is most similar to an appearance in the far past.

Appearance-Based Experience Selection:

This experience selection method operates by determining a relevant subset of experiences with similar appearance to the *live view*. This Bag of Words (BoW) experience selector first published in MacTavish et al. (2017), selects a small subset of experiences based on a BoW appearance summary, illustrated in Figure 5.4.

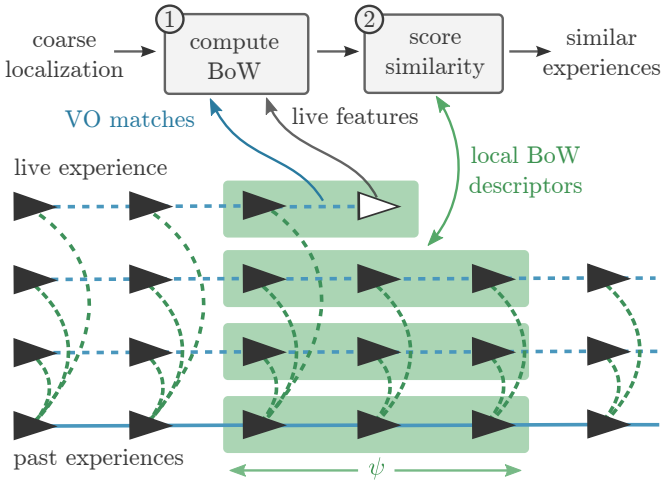


Figure 5.4: An overview of the experience selection algorithm using BoW. 1. Given the VO feature matches for the live frame, a sliding-window BoW descriptor is constructed. 2. The BoW descriptor is compared against those for past experiences, centered on the coarse localization. The most similar experiences are selected for use in the remainder of the localization problem.

To support this method, each vertex in the STPG contains a BoW descriptor, quantized from the stable (tracked by VO) features in the keyframe against a local visual vocabulary. At the start of each MEL problem, the local BoW descriptor (green rectangle) around the live vertex (white triangle) is

compared to those of each past experience (lower green rectangles) using the cosine similarity. The N experiences with the highest similarity to the live experience are selected for MEL, where N is chosen to maintain real-time performance (5-10). Since this comparison is very fast, this method allows us to triage a large number (100s) of experiences very quickly.

Collaborative Filtering Experience Selection

The latest experience selection method enabled in the VT&R 2.0 system is the Collaborative Filtering (CF)-based selector. CF is a recommender system, which recommends items based on a history of users expressing preference for items, when little else is known about the items (Liu et al., 2010). This method originally intended for use cases such as Netflix movie recommendations has been adapted for use in the VT&R 2.0 system by treating users as experiences and items as matched map landmarks. The intuition behind this method is that experiences with similar landmarks matches to the live view will be recommended first. This research is still ongoing with plans for a colleague to publish the full algorithm in Mactavish et al. (2018).

5.2.6 Pipeline Parallelization

Parallelization of the state estimation pipeline is a crucial requirement to run the VT&R 2.0 system fast enough where it can operate online and support realistic vehicle speeds. This parallelized pipeline, shown in Figure 5.5, is an extension of the VO threads described in Section 4.3.2, with the ability to provide multi-experience localization. The navigation system consists of two primary estimation threads:

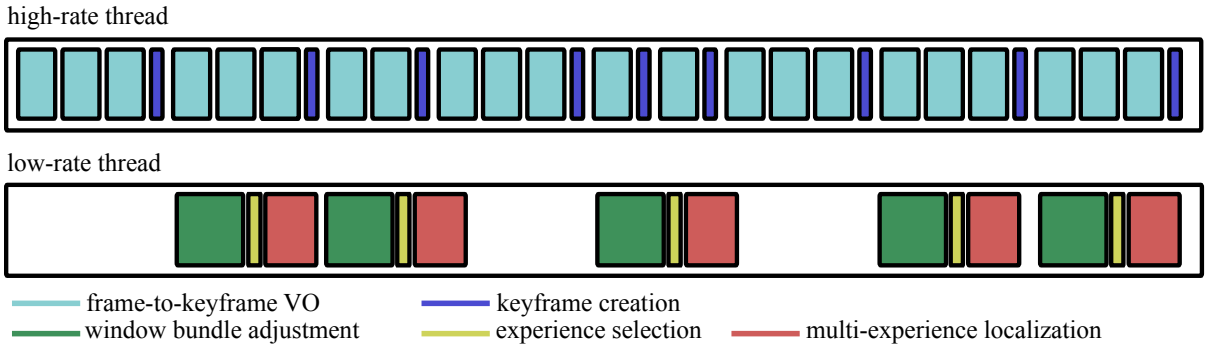


Figure 5.5: Overview of the parallel state estimation threads used in the VT&R 2.0 system. *top:* The high-rate estimation thread computes vehicle motion estimates through the frame-to-keyframe VO pipeline (Section 4.3.2) and creates vertices in the live experience which triggers estimation in the parallel, low-rate thread. If an estimation task is already underway in the low-rate thread at the time of keyframe creation, then its low-rate estimation is skipped. *bottom:* The low-rate estimation thread refines the transforms and landmarks in a sliding window containing the most recent vertices in the live experience (Section 4.3.2), selects a fixed amount of experiences to localize against (Section 5.2.5), and computes localization between the live and privileged experience using the MEL algorithm (Section 4.3).

a high-rate thread that runs at the frame rate of the sensor, and a low-rate thread that roughly runs at the rate of keyframe creation. In the high-rate thread, the frame-to-keyframe VO pipeline described in Section 4.3.2 is run at the frame rate of the sensor. At the conclusion of each frame-to-keyframe solve, a decision is made whether or not to create a new vertex in the live experience based on distance traveled and the number of inliers matched to the previous keyframe. If a keyframe is created, then the low-rate thread spawns a windowed VO bundle adjustment problem, described in Section 4.3.2, on

the latest vertices in the graph. After bundle adjustment is completed, the low-rate thread selects the subset of experiences to localize against, as per Section 5.2.5, and then performs localization between the newly created live vertex and closest vertex in the privileged experience using the MEL algorithm. In the meantime, the high-rate thread continues to estimate the rover's motion and provides updates to the path tracking controller while autonomously following a path.

If a decision is made to create a vertex while the low-rate thread is still busy refining and localizing the previous vertex, the new vertex is created with its localization set by integrating VO, and the low-frame rate job for the new vertex is skipped. We firmly believe that while running online, it is preferable to have higher quality bundle adjustment and localization at a slower rate rather than sacrificing localization computation time to provide faster state estimates.

5.3 Field Tests

This section details the autonomous path-following field tests that were devised to experimentally validate the VT&R 2.0 system. the field tests were conducted to demonstrate the VT&R 2.0 system's ability to autonomously navigate networks of paths in challenging outdoor environments over long time periods.

Table 5.1: Overview of the VT&R 2.0 vision-in-the-loop Field Tests

Field Test	Location	Dates	Auto. Dist.	Auto. Rate
Ethier Gravel Pit	Sudbury, ON	06/10/16 - 06/15/16	140.0 km	99.59%
UTIAS In The Dark	Toronto, ON	07/19/16 - 07/20/16	10.5 km	100.00%
UTIAS Multi-Season	Toronto, ON	01/31/17 - 05/23/17	28.0 km	99.98%

An overview of the VT&R 2.0 vision-in-the-loop field tests is presented in Table 5.1. In total, over 178 km of vision-in-the-loop autonomous path following was conducted across three distinct field tests. The first field test, detailed in Section 5.3.2, demonstrates the system's ability to navigate across a large-scale network of paths over short-term appearance change due to lighting and weather. The second field test, detailed in Section 5.3.3, demonstrates the system's ability to handle a large amount of experiences on one path and to operate at night using a path demonstrated during the day. Through the use of on-board headlights, and the ability to bridge the appearance gap with the MEL algorithm, the VT&R 2.0 system is capable of autonomous traversal over 24-hour periods in outdoor environments. The final field test, detailed in Section 5.3.4, demonstrates the VT&R 2.0 system's ability to autonomously navigate for long time periods across difficult seasonal changes. In this field test a 160 m loop is autonomously repeated for over 100 days between the months of February and May, experiencing appearance change as extreme as winter vs. summer.

5.3.1 Hardware

The hardware configuration for used for all field tests in this section is displayed in Figure 5.6. A Clearpath Robotics Grizzly RUV serves as our mobile robot platform and is equipped with a payload that includes a suite of interoceptive and exteroceptive sensors. For all field tests, the only sensor used for localization and mapping was the forward facing PGR Bumblebee XB3 stereo camera labeled in

Figure 5.6: Clearpath Grizzly RUV and its sensor configuration. The robot is equipped with a forward- and rear-facing PGR Bumblebee XB3 cameras, a GPS receiver, and four 9W LED headlights for night-time operation. The robot contains a ROS enabled embedded computer that controls its motors and safety monitors.

Figure 5.6. During the field tests, GPS data was collected for the purpose of visualization only. All of the VT&R 2.0 code either ran on a a Lenovo W540 laptop with a Intel® Core™ i7-4800MQ CPU or a Lenovo P50 laptop with a Intel® Core™ i7-6820HQ CPU. Amount of available on-board RAM for each trial was set to either 16GB or 32GB. Hard drive space on the laptops was limited to 1TB. For night-time driving, a pair of forward-facing LED headlights were used to illuminate the scene.

5.3.2 Ethier Gravel Pit

Between the dates of 10/06/2016 and 16/06/2016, an extended field test of VT&R 2.0 was conducted at an untended gravel pit at Ethier Sand & Gravel in Sudbury, Canada. This field test was designed to stress-test our system's ability to traverse a large-scale network of paths, in an unstructured environment, for an extended period of time, over significant intra-seasonal appearance change due to lighting and weather. This location, shown in Figure 5.7, was selected for its vast scale and variety of challenging environment. Examples of the difficult areas of the pit covered in this field test include: i) a vegetation-rich ridge containing tall grass and a strong tree line, ii) long stretches of gravel road with flanked by tall trees which sway in the wind and provide strong shadows on sunny days, and iii) open stretches of desert areas with shifting sand that provide little visual texture.

To test our system's ability to navigate in this difficult terrain, the 5 km network of connected paths shown in Figure 5.7 was manually demonstrated to the robot and autonomously traversed for 10 days, totalling over 140 km of autonomous navigation. Daily field test activities are outlined in Table 5.2. The majority of the network was taught on the first three days of testing during overcast conditions, with < 1 km of additions being added on subsequent days. During the first five days, the network was traversed from day to night, accumulating over 95 km of driving. On each of the remaining six days, the robot autonomously traversed between 5 and 10 km a day. For each day, the autonomy rate remained

above 99%. It is interesting to note that the majority of manual interventions occurred on the first two sunny days. Details on the causes of manual interventions are left for Section 5.5.1.

Table 5.2: Overview of the 2016 Ethier gravel pit field test

Date	Start [hh:mm]	End [hh:mm]	Weather	Teach Dist. [km]	Auto. Dist. [km]	Intervention Dist. [m]	Auto. Rate
06/06	11:15	21:19	Rainy	1.56	13.45	12.56	99.91
06/07	05:42	20:42	Overcast	1.88	17.61	8.74	99.95
06/08	07:21	21:10	Rainy	0.83	23.95	72.66	99.70
06/09	06:25	21:23	Sunny	0.03	19.10	148.69	99.22
06/10	07:13	21:17	Sunny	0.33	20.76	171.62	99.17
06/11	07:40	20:46	Sunny	0.22	09.64	41.49	99.57
06/12	09:59	22:35	Sunny	0.20	08.95	20.04	99.78
06/13	10:38	22:45	Sunny	0.00	07.87	27.65	99.65
06/14	07:33	22:50	Sunny	0.00	07.63	53.42	99.30
06/15	12:16	17:57	Sunny	0.00	07.37	2.27	99.97
06/16	09:16	14:57	Sunny	0.00	04.15	2.85	99.93
Total:	—	—	—	5.0	140.5	561.99	99.60

During the 10 day field test, the VT&R 2.0 system experienced a significant amount of variance in the appearance of the environment due to intra-seasonal effects from lighting and weather. Examples of this appearance change seen during the field test for three distinct regions (rows in the figure) of the network is displayed in Figure 5.8. The majority of the network was taught in overcast conditions, this is reflected in the left-hand image of every row. The appearance varied significantly due to lighting, especially during sunrise and sunset when the sun is low on the horizon, often causing sun glare in the images. This effect can be seen in Figure 5.8a, where the sun was low on the horizon during sunset and oversaturated a large amount of the image. Through the use of on-board headlights, the VT&R 2.0 system continued operation at night, in the total absence of ambient light. This can be seen in the fourth image of Figure 5.8b. Another interesting source of appearance change is terrain modification from vehicles. This can be seen in Figure 5.8c, where the path cuts through an open desert area that was adjacent to the main operation site of a separate field trial. Due to the high traffic of vehicles in this area, the sand continuously changed appearance, proving to be one of the most challenging areas of the field test, causing more than one manual intervention. Despite these drastic changes in appearance, the VT&R 2.0 system successfully localized all images in these figures to their respective privileged experiences using the MEL algorithm. Results from this field test are reported in Section 5.5.1 and the few failure points encountered in the field are discussed in Section 5.5.1.

5.3.3 UTIAS In The Dark

The Ethier field test demonstrated the VT&R 2.0 system's ability to navigate large-scale networks across intra-seasonal appearance change. However, the size of the Ethier network meant that the amount of experiences gathered on any path was relatively small. In order to demonstrate the system's ability to handle a large amount of experiences while performing online localization, we conducted a simple field test at the University of Toronto Institute for Aerospace Studies (UTIAS) campus. Shown in Figure 5.9, this field test consisted of demonstrating a 250 m loop and autonomously repeating the loop many times, including a continuous 24-hour period with the aid of on-board headlights. This field test

Figure 5.7: Orthomosaic imagery of the 5 km network of paths at the Ethier Sand and Gravel in Sudbury, ON. This network was taught during a 10 day field trial to experimentally validate the VT&R 2.0 system's ability to perform long-range, long-term vision-based autonomous path following in difficult environments, highlighted in the image. Over the course of the field test, the robot autonomously traversed 140 km with an autonomy rate of 99.59% of distance traveled.

Table 5.3: Overview of the autonomous traverses conducted during the UTIAS In the dark field test. The column Δt corresponds to the duration between the teach run and the repeat run.

	ID	Start Time	Duration [hh:mm]	Δt [hh:mm]	Autonomy [%]	Distance [m]
Day	0	16/07/19 09:02	00:05	00:08	100.0	136.60
	1	16/07/19 09:09	00:10	00:14	100.0	249.80
	2	16/07/19 09:46	00:11	00:51	100.0	249.47
	3	16/07/19 10:58	00:10	02:03	100.0	249.75
	4	16/07/19 12:29	00:10	03:34	100.0	249.77
	5	16/07/19 13:57	00:10	05:03	100.0	249.41
	6	16/07/19 15:24	00:10	06:29	100.0	249.26
	7	16/07/19 16:48	00:10	07:54	100.0	249.19
	8	16/07/19 18:23	00:11	09:28	100.0	248.72
	9	16/07/19 18:36	00:10	09:42	100.0	247.98
	10	16/07/19 19:31	00:12	10:37	100.0	246.37
	11	16/07/19 20:01	00:13	11:07	100.0	249.80
	12	16/07/19 20:17	00:10	11:23	100.0	249.79
	13	16/07/19 20:30	00:11	11:36	100.0	249.31
	14	16/07/19 20:44	00:15	11:50	100.0	249.49
Sunset	15	16/07/19 21:00	00:10	12:06	100.0	249.15
	16	16/07/19 21:12	00:13	12:18	100.0	249.34
	17	16/07/19 21:26	00:10	12:32	100.0	249.34
Night	18	16/07/19 21:36	00:10	12:42	100.0	250.07
	19	16/07/19 21:48	00:10	12:53	100.0	250.47
	20	16/07/19 21:58	00:10	13:04	100.0	251.29
	21	16/07/19 22:28	00:11	13:33	100.0	250.25
	22	16/07/19 22:40	00:11	13:45	100.0	250.75
	23	16/07/20 04:29	00:12	19:35	100.0	251.26
	24	16/07/20 05:10	00:11	20:16	100.0	250.75
	25	16/07/20 05:22	00:10	20:28	100.0	249.18
	26	16/07/20 05:34	00:11	20:40	100.0	248.69
	27	16/07/20 05:48	00:14	20:54	100.0	249.19
Sunrise	28	16/07/20 06:04	00:10	21:10	100.0	249.41
	29	16/07/20 06:16	00:12	21:22	100.0	249.78
	30	16/07/20 06:31	00:11	21:37	100.0	252.60
	31	16/07/20 06:44	00:11	21:50	100.0	249.35
Day	32	16/07/20 07:03	00:10	22:09	100.0	249.63
	33	16/07/20 07:19	00:11	22:25	100.0	249.91
	34	16/07/20 07:39	00:12	22:45	100.0	249.51
	35	16/07/20 08:35	00:10	23:40	100.0	249.96
	36	16/07/20 09:18	00:11	24:24	100.0	249.52
	37	16/07/20 10:21	00:10	25:27	100.0	249.42
	38	16/07/20 11:26	00:10	26:32	100.0	249.34
	39	16/07/20 12:29	00:11	27:34	100.0	249.79
	40	16/07/20 14:01	00:10	29:07	100.0	249.24
	41	16/07/20 15:30	00:10	30:36	100.0	250.38
Total:	43	—	7h 55m	—	—	10.5 km

(a) Examples of appearance change observed in a partially vegetated desert area.

(b) Examples of appearance change observed in a vegetation rich area.

(c) Examples of appearance change observed in an open desert area.

Figure 5.8: Examples of the appearance change observed in different areas of the 5 km network of paths autonomously traversed during the Ethier Gravel Pit Field test. Each row of the figure shows an example image from the privileged map on the left and images of the same place under varying appearance while the robot was autonomously traversing the path. In all cases the MEL algorithm was able to successfully localize the views with respect to the map image.

demonstrates the system's ability to both operate over 24-hour periods outdoors in the absence of light and handle a large amount of experiences in the map through the use of the system's appearance-based BoW experience selector discussed in Section 5.2.5. We note once again that experience selection is not a novel contribution of this thesis, so the analysis of the field test results focuses on the performance of the MEL localization system given the experiences provided by the selector.

Activities of the field test are overviewed in Table 5.3. The test begins with a manual demonstration of the 250 m loop at 09:02. The robot then proceeds to autonomously repeat the path every hour and a half from 09:09 to 22:40, with on-board headlights illuminating the scene after sunset. The robot resumes traversal of the path before sunrise at 04:29, traversing approximately every hour until 15:10. On both days of the field test the sky condition was sunny, allowing the algorithm to experience every variation of lighting seen over a diurnal cycle. This can be seen in Figure 5.10, which shows the changing appearance of the scene over the 30 hour period. Notable appearances caused by lighting are long shadows cast at sunset, illumination by on-board headlights at night, and harsh sunglare occurring during sunrise. Despite these challenges, the MEL algorithm in conjunction with the BoW experience selector was able to autonomously traverse the path 42 times accumulating over 10 km of driving with an autonomy rate of 100.0%. Detailed results of this field test related to the performance of the MEL algorithm are detailed in Section 5.5.2.

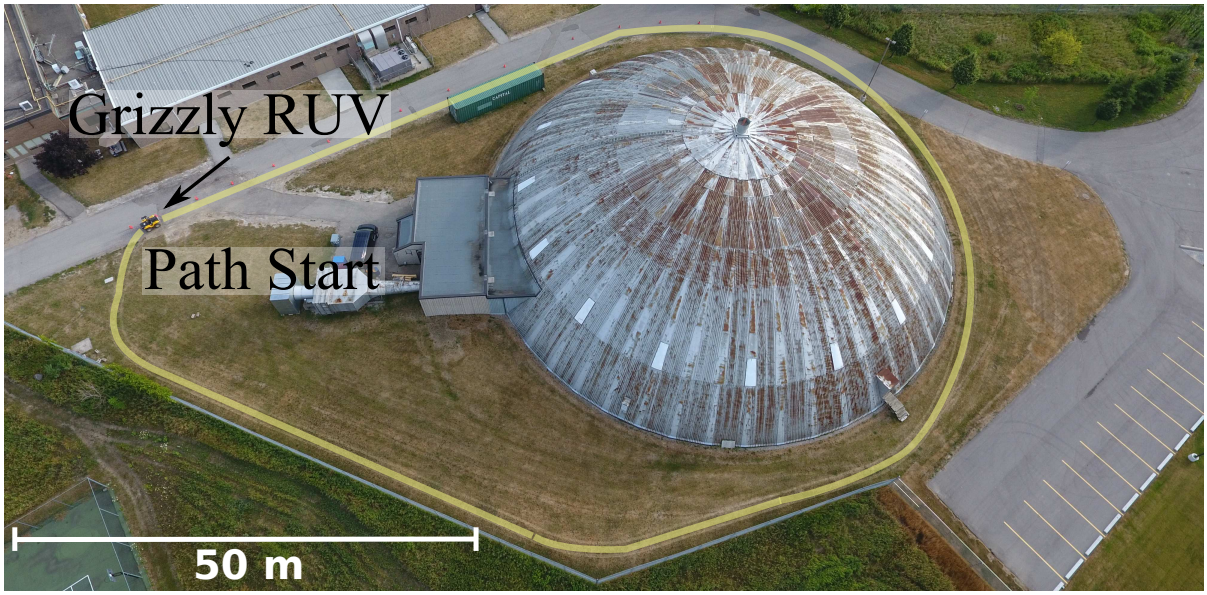


Figure 5.9: Overview of the experiment grounds, the grizzly RUV is show at the start of the approximately 250m path.

Figure 5.10: Examples of the appearance change observed during the 30-hour UTIAS In the Dark field test. This short field test demonstrates the VT&R 2.0 system's ability to handle man experiences of a single scene in one map (40+) as well as operate continuously over a 24-hour period with on-board headlights illuminating the scene at night.

5.3.4 UTIAS Multi Season

The previous VT&R 2.0 field deployments demonstrated the system's ability to autonomously navigate large-scale networks of paths across short-term appearance change due to lighting, but did not show the system's ability to navigate across long-term, seasonal appearance change. Accordingly, a simple field test was devised to demonstrate the VT&R 2.0 system's capability to bridge the gap across the extreme appearance change observed as the winter season transitions into summer in Canada.

On the date of January 31st, 2017, a 165 m loop, shown in Figure 5.11, was manually taught at the University of Toronto Institute for Aerospace Studies (UTIAS). This semi-structured environment consists of an open meadow with short and tall grass, sparse shrubs, a perimeter fence, and a tennis court. Over the next 17 weeks, the robot autonomously traversed the loop 163 times, totaling over 28 km of driving with an autonomy rate of 99.98% of distance traveled. An overview of the field test activity can be seen in Table 5.4 and examples of the appearance change experienced during the test can be seen in Figure 5.12.

In the first three weeks of testing, 01/29-02/18, 74 of the 175 autonomous repeats were conducted. This disproportionately large amount of activity was to ensure that the MEL algorithm sufficiently

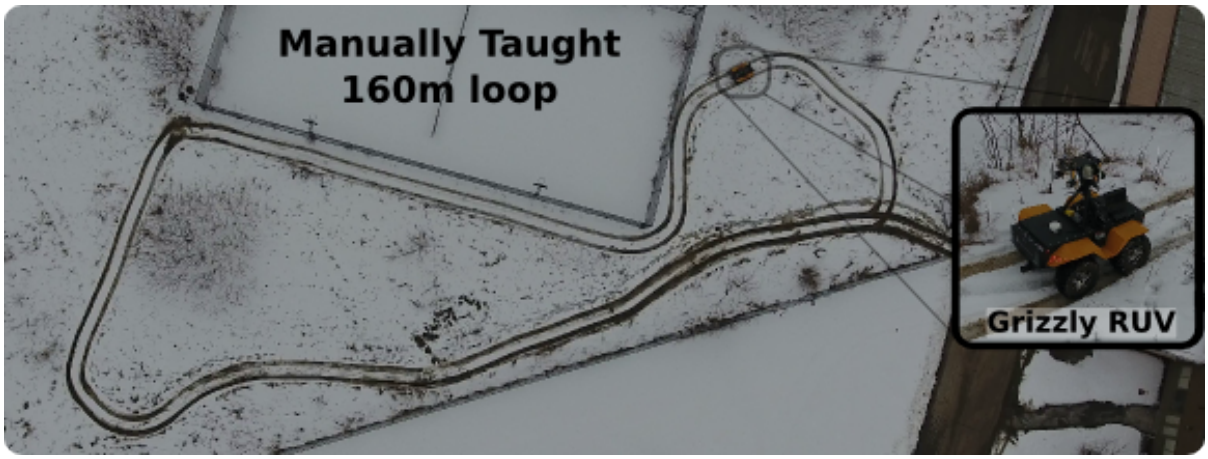


Figure 5.11: A Grizzly RUV deployed with the VT&R 2.0 system navigating a small 160 m loop at the University of Toronto Institute for Aerospace Studies (UTIAS) during the 2017 multi-season field test. This field test demonstrates the VT&R 2.0 system's ability to autonomously operate over long time periods. Over the course of four months the robot autonomously repeated this loop 163 times experiencing appearance change as extreme as snow, freezing rain, and vegetation growth.

captured the rapid appearance change seen at the tail end of winter. This winter appearance change can be observed in the first three rows of Figure 5.12, where the appearance fluctuated daily between light snowfall, freezing rain, heavy snow fall, and snow melt. The appearance of the scene between the weeks of 02/19 and 04/01 was fairly constant, consisting of dead vegetation with the occasional light snowfall and melt. During this six week period, the robot autonomously traversed the loop 47 times. For the remainder of the field test, the appearance of the scene gradually transitioned from a meadow of dead, flattened vegetation to the meadow of tall grass seen in the last image of Figure 5.12. During this time of gradual appearance change, the robot autonomously traversed the loop 58 times until the conclusion of the field test on the 17th week ending on 05/27 when the system failed to localize due to the rapidly growing grass. Tall grass in particular is challenging to vision system's for two reasons: i) during the transition from spring to summer it rapidly grows in height, and ii) on windy days the grass sways in the wind, breaking geometric consistency checks in both VO and localization. A discussion on this issue can be found in Section 5.5.3.

Despite these challenges, we demonstrate with this field test a significant increase in robustness to changes in long-term seasonal appearance for vision-based navigation systems that require precise, metric localization. We furthermore show that the challenges for vision-based systems presented by winter environments are mitigated through the use of our multi-experience localization framework. Detailed results for this field test are presented in Section 5.5.3, with a discussion on the challenges and failure points encountered during this test in Section 5.5.3.

Experience Selection Over the course of the field test, the VT&R 2.0 system made use of multiple experience selection algorithms. This is due to the nature of conducting long-term field tests on methods that are in the process of being researched. The experience selector's used during this field test are overviewed in Table 5.4. A high-level overview of each selector is presented in Section 5.2.5, and is reiterated here that they are not a novel contribution of this thesis, rather they are work being done in conjunction with this thesis. During the first two months of the field test, the ToD experience selector

Table 5.4: Overview of the 2017 UTIAS multi-season field test

Week	Conditions	No. of Repeats	Exp. Selector
01/29 - 02/04	Light Snow	18	ToD
02/05 - 02/11	Freezing Rain, Snow Melt	27	ToD
02/12 - 02/18	Heavy Snow, Snow Melt	29	ToD
02/19 - 02/25	Sunshine	9	ToD
02/26 - 03/04	Overcast	6	ToD
03/05 - 03/11	Overcast	5	ToD
03/12 - 03/18	Snow fall, Snow Melt, Sunshine	14	CF
03/19 - 03/25	Snow fall, Snow Melt, Sunshine	6	CF
03/26 - 04/01	Overcast, Sunshine	7	CF
04/02 - 04/08	Snow Fall, Snow Melt, Grass Growth	7	CF
04/09 - 04/15	Sunshine, Grass Growth	6	CF
04/16 - 04/22	Sunshine, Grass Growth	10	CF
04/23 - 04/29	Sunshine, Grass Growth	5	CF
04/30 - 05/06	Sunshine, Grass Growth	5	CF
05/07 - 05/13	rain, Grass Growth	10	CF
05/14 - 05/20	rain, Grass Growth	13	CF
05/21 - 05/27	Sunshine, Grass Growth	3	CF

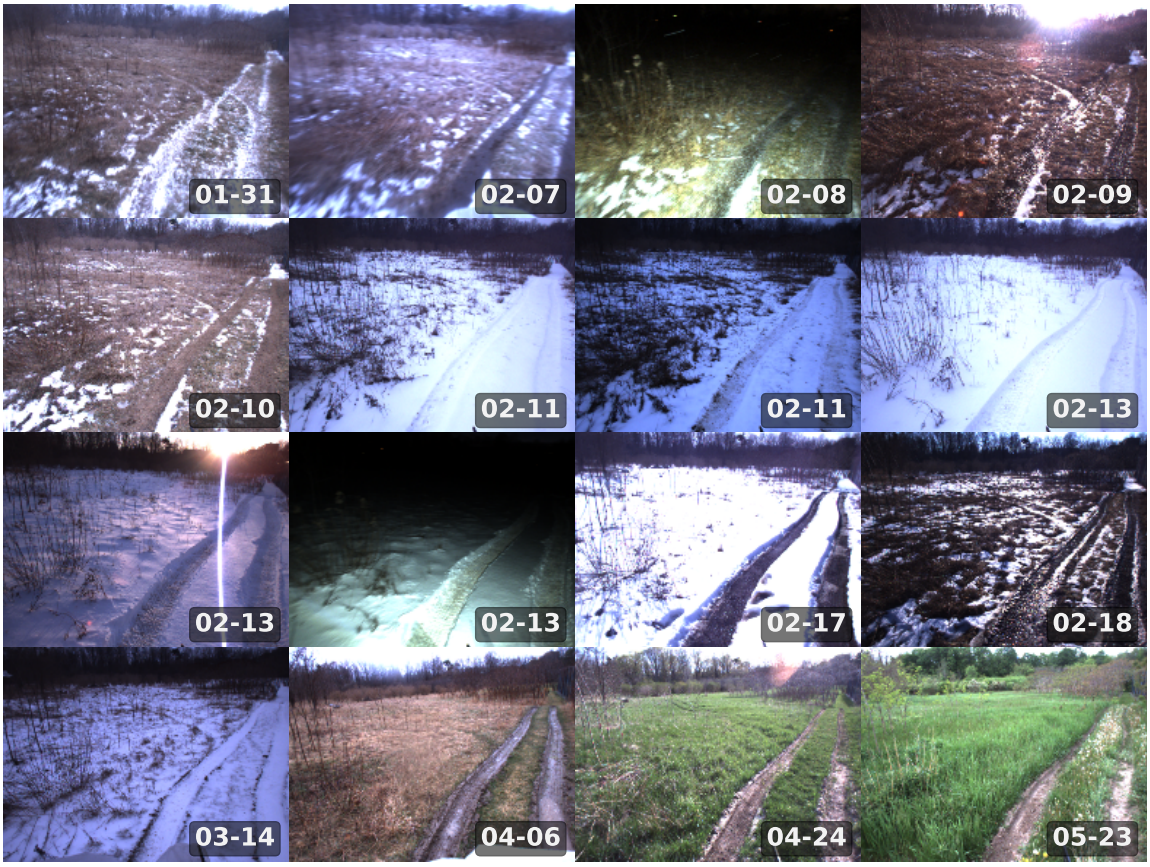


Figure 5.12: Examples of appearance change primarily due to winter weather observed over three months while autonomously following a path at the University of Toronto in February, 2017.

was chosen with a fixed set of 15 experiences chosen for each localization problem. This naive solution to selecting experiences proved to be sufficient for autonomous traversal when the appearance of the scene gradually changes. However, for rapid appearance change due to snowfall and snow melt, the appearance of the scene can more closely resemble experiences from weeks in the past. To test the ability for the VT&R 2.0 system to perform "temporal loop closures", manual experiences that were the most similar to the live experience were hand picked after snowfall on the date of 02/19. For the remainder of the field test, the latest experience selection algorithm, Collaborative Filtering (CF) was used to select experiences.

5.4 Evaluation Metrics

This section provides details on the metrics used to quantify the performance of the VT&R 2.0 system with respect to the field tests and offline experiments provided in the previous section. To quantify this performance we selected five quantitative metrics: 1. Cross-track uncertainty, 2. Distance Driven on Dead Reckoning, 3. Feature inlier count, 4. Autonomy Rate, and 5. Computation time. We provide details on each metric and provide example figures explaining how they are represented in the results section.

5.4.1 Cross-track uncertainty

This is our primary metric for judging localization success. We define cross-track uncertainty as the one-standard-deviation uncertainty of our lateral translation estimate relative to the privileged path. This tells us how uncertain we are to the left or the right while following the privileged path, and can be directly interfaced with our path-tracking controller to provide safe autonomous driving based on the lateral constraints of the path. To visualize this metric in the autonomous path-following field test results, we display the Cumulative Distribution Function (CDF) of the $1-\sigma$ cross-track uncertainty for each autonomous traverse of a given path. An example of this visualization is displayed on the left side of Figure 5.13, where each line displays statistics for a full autonomous traverse colored by the number of days since the privileged experience in the map was created. This metric can be intuitively read as: "for y% of the traverse, the robot incurred less than x m of $1-sigma$ cross-track uncertainty".

It is important to note when considering coss-track uncertainty as a metric for localization success that while uncertainty is calculated at every stage of the algorithm from keypoint detection to landmark transformation, a rigorous evaluation of our uncertainty estimates with respect to ground truth to ensure consistency had not yet been established during when the field tests were conducted. Therefore we treat this metric as a way to compare relative performance between experiments and do not necessarily trust the exact scale of our uncertainty estimates.

5.4.2 Distance Driven on Dead Reckoning

Another metric for judging the quality of localization is the distance the robot drove on dead reckoning during an autonomous traverse. As discussed in Section 5.2.6, localization to the privileged path is estimated when a keyframe is added to the live experience. In a nominal traverse, the robot is only driving on dead reckoning between keyframes, with regular localization updates being made at the rate of keyframe creation. When localization begins to fail, usually due to a lack of data correspondences

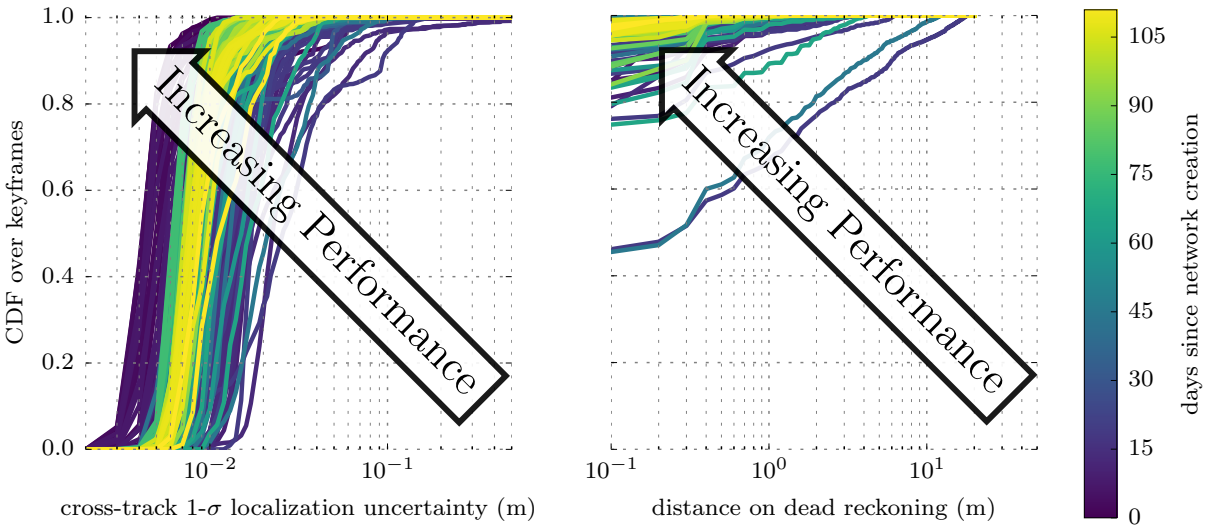


Figure 5.13: Example image of the visualization of cross-track uncertainty for every autonomous traverse.

between the live view and the experiences in the map, the estimate of the robot state is integrated from the last known localization using VO to provide an estimate to the path tracker. This metric analyzes how often this occurs during an autonomous traverse. As this distance on dead reckoning increases, so does the uncertainty of its localization estimate. Therefore, this metric is highly correlated with the cross-track uncertainty metric. This can be seen in the example visualization of this metric on the right side of Figure 5.13. Like cross-track uncertainty, distance on dead reckoning is visualized as the CDF over each autonomous traverse.

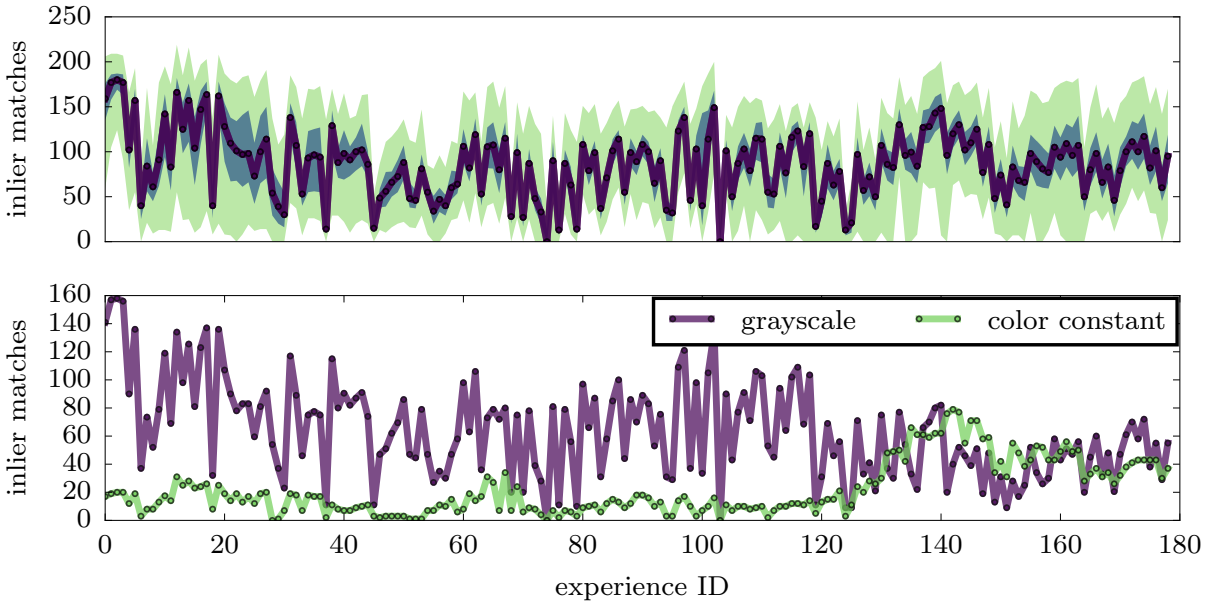


Figure 5.14: Example image of the visualization of cross-track uncertainty for every autonomous traverse.

5.4.3 Feature inlier count

This metric is simply the median number of inliers observed (after RANSAC) over all channels (e.g. grayscale and color constant) over all localization estimates for each autonomous traverse. As the VT&R 2.0 localization system uses multi-channel state estimation as described in Chapter 3, inlier matches are obtained by summing the amount from all channels. In the VT&R 2.0 field test two information channels were used: grayscale and color-constant stereo images. An example of how this metric is visualized for an autonomous path-following field test is seen in Figure 5.14. The total amount of inler matches from *all* channels is shown in the top of Figure 5.14. In this figure, the median inler match count is displayed as a dark purple line, with the dark shaded area representing the upper and lower quartile, and the light shaded area representing the minimum and maximum inlier values of the distribution. A breakdown of median inlier matches from each information channel is shown in the bottom figure.

5.4.4 Autonomy Rate

This metric is simply the percentage of a traverse by distance traveled where the robot remained autonomous. This number decreases from 100% when manual interventions are required to continue operation. Manual interventions typically occur when the localization to the privileged path is poor, and the robot is out of its tracks. While the autonomy rate is very high for all field tests, there are certain situations that remain challenging for the VT&R 2.0 system. Autonomy rate is reported for each field test, with a detailed discussion on when the system required manual interventions in Section 5.6.

5.4.5 Computation time

As complexity of the MEL algorithm scales linearly with the number of experiences in the worst case scenario (when matching reaches the upper bound of time), we are interested in observing the average computation time of localization for each autonomous traverse. Similar to the localization inlier metric, we present the median computation time along with the upper and lower quartile, and the min/max. In order for this algorithm to support vision-in-the-loop route following, this solve time needs to be fast enough to support the incoming localization requests from the path tracker.

5.5 Results

The goal of this section is to demonstrate the VT&R 2.0 system's ability to perform long-term, vision-based autonomous path following using the MEL algorithm. In particular, we aim to provide quantitative results that demonstrate the ability to perform metric localization across long-term appearance change using multiple experiences. This is achieved through the analysis of the performance of the VT&R 2.0 system while conducting the experiments and field tests outlined in Section 5.3, with respect to the metrics described in Section 5.4.

5.5.1 Ethier Gravel Pit

The results of Chapter 4 showed that the MEL algorithm, in an offline setting, is capable of providing robust metric localization across significant appearance change due to lighting, with a small amount of bridging experiences. This section presents results on the Ethier Gravel Pit field test, introduced in Section 5.3.2. These results show that the MEL algorithm is capable of providing vision-in-the-loop navigation to a large-scale autonomous path-following system in unstructured outdoor environments over short time periods. We show VT&R 2.0 field results on more than 140 km of autonomous driving across 10 days with an autonomy rate of 99.59% of distance traveled. Furthermore, through detailed results on specific sections of the network we show that the performance vision-based navigation is still dependent on the environment of the scene.

Summary Results

This section provides results summarizing localization performance across all 646 autonomous traverses conducted during this field test, totalling 140 km of autonomous driving.

Autonomy Rate Figure 5.15 visualizes the autonomy rate and manual interventions required during the field test due to localization failures. The left-hand plot shows the manual interventions as a function of distance traveled and the right-hand plot shows the distribution of manual interventions across the network. Interventions are categorized into three types: trivial, minor, and major.

Trivial interventions (blue dots) were manifestations of a software bug that occurred when the system experienced a VO error, which would stop the robot until the operator drove it forward approximately 0.3 m to trigger a new vertex in the graph[1]. As the nature of this error is somewhat random, the distribution of trivial interventions is nearly uniform across the network. Minor interventions (green dots) consisted of small course corrections when the robot was slightly out of its tracks to continue autonomous operation. These were results of poor localization for short distances, which correspond to areas of the network that are challenging for vision-based navigation. Major interventions (yellow lines) occurred when the robot was significantly off course and could not recover. These are a result of extended localization failures without the ability to recover. Out of the 646 autonomous traverses conducted during this field test, the system experienced seventeen minor interventions and five major interventions. Looking at Figure 5.15, it can be seen that the robot experienced major interventions in the following areas: i) the tree-lined road when it was sunny and windy (top left), ii) the vegetation-rich area during high winds (mid right), and iii) the open desert area in sunny conditions after heavy vehicle

[1]This bug was resolved directly after this field test and is not present in other field tests detailed in this chapter.

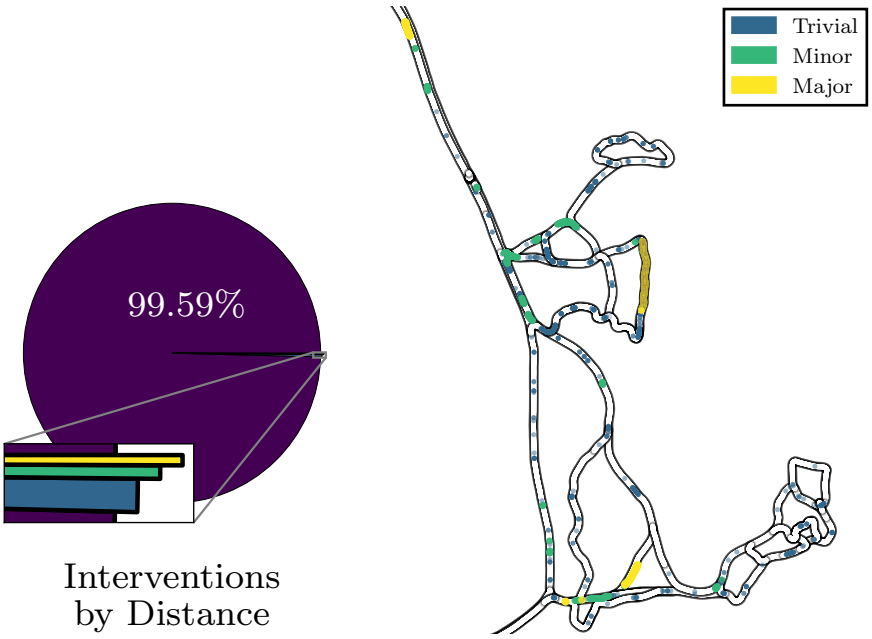


Figure 5.15: Visualization of autonomy rate and manual interventions during the 140 km field trial due to localization issues. Interventions are categorized by the following severities: i) fully autonomous, no interventions, ii) trivial interventions due to software bugs, iii) minor interventions to steer the robot back on course. iii) major interventions due to severe localization failures. *left:* Autonomy rate as a function of distance traveled. Of the 140 km of distance traveled during the field trial, 99.59% of that distance was fully autonomous. *right:* Intervention occurrences for all autonomous runs plotted on the network. Major interventions occurred in areas that remain difficult for vision-based navigation and is discussed in further detail in Section 5.5.1.

traffic. Despite these difficult conditions, the robot maintained an autonomy rate of 99.59% of distance traveled. A discussion on vision-based localization in these difficult areas is reserved for Section 5.5.1.

Cross-Track Uncertainty Results of the field test with respect to cross-track uncertainty are displayed in Figure 5.16. The left-hand figure shows the CDF of the $1-\sigma$ cross-track uncertainty for all 646 autonomous traverses of the field test. These results show that for the majority of autonomous traverses, the cross-track uncertainty remained below 1 m for 100% of each traverse and below 0.1 m for 50% of each traverse. The traverses with higher uncertainties (dark blue lines) occurred during the 3rd and 4th day of testing, the first days of bright sunshine. It can be seen in Table 5.1, that these first sunny days correspond to the majority of manual interventions during the field test. This is partially due to sunny conditions causing sun glare during sunrise and sunset and a few key areas of the network that are especially difficult for vision-based systems. A detailed analysis on the causes behind these difficult traverses is reserved for a discussion of the challenges and lessons learned in Section 5.6.

Distance on Dead Reckoning Results of the field test with respect to distance driven on dead reckoning are displayed in Figure 5.16. The right-hand figure shows the CDF of the distance driven on dead reckoning for all 646 autonomous traverses of the field test. These results show that the distance driven on dead reckoning for each run is highly correlated to the cross-track uncertainty observed during that run. For the majority ($> 98\%$) of autonomous traverses, the robot drove less than 20 m on dead reckoning for 100% of each respective traverse's distance. For the other 2%, all but one drove less than 35 m on dead reckoning. The one outlier exceeded 60 m on dead reckoning and occurred in an

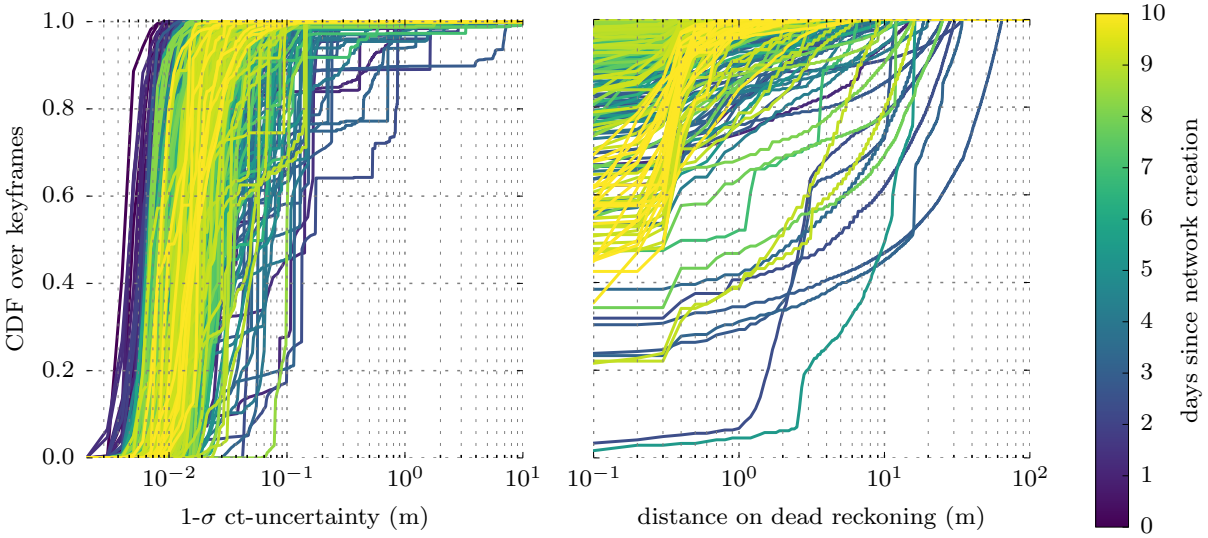


Figure 5.16: Localization Results for all 646 autonomous traverses of the Ethier Gravel Pit field test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses.

open-desert area of the field test where there are few stable features in the field-of-view of the stereo camera. A detailed analysis on the causes behind these difficult traverses is reserved for a discussion of the challenges and lessons learned in Section 5.6.

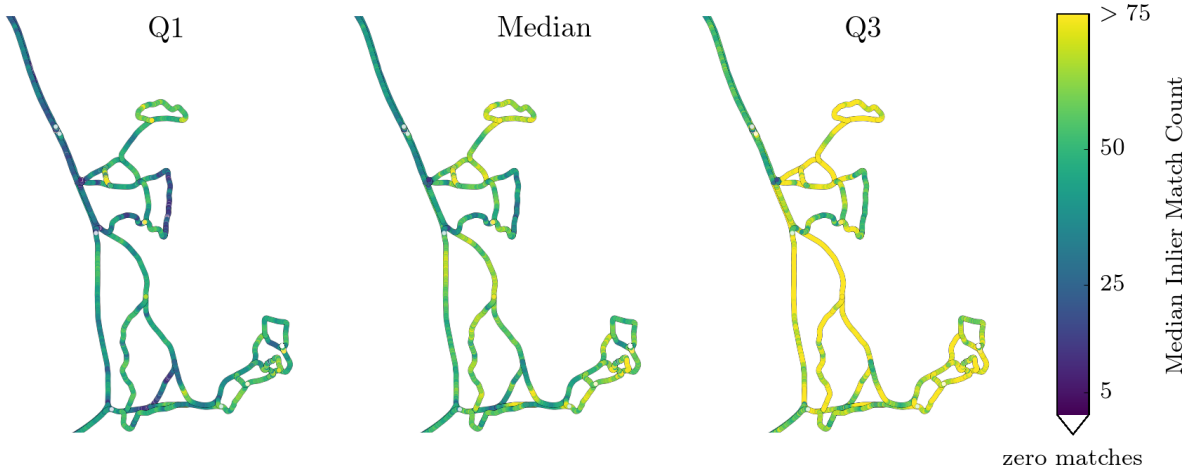


Figure 5.17: Median inlier matches over the entire network for every autonomous traverse. Each vertex in the privileged, manually driven network is colored by the median amount of inlier matches observed by all spatially adjacent vertices in autonomous experiences. It is interesting to note that there are certain areas of the network with much lower median match counts.

Inlier Feature Matches Results of the field test with respect to inlier feature matches are displayed in Figure 5.17 and Figure 5.18, which shows the distribution of inlier matches across the entire network and detailed statistics for each autonomous traverse, respectively. Figure 5.17 illustrates the localization inlier feature match counts for every autonomous traverse. To summarize localization performance, each

privileged vertex in the graph is colored by the median inlier match count and the upper/lower quartile (Q1/Q3) of *all* adjacent autonomous vertices. It can be seen that for most of the network, this median value is at least 50 inlier matches, with some areas having match count of over 75. However, there are areas of the network with lower counts in the range of 10-20 matches. These regions tend to contain areas of ephemeral ground features. Examples include: i) tall grass on windy days, ii) tree lines with strong shadows, and iii) sandy areas with concentrated vehicle traffic. While the MEL algorithm can bridge extreme appearance gaps, there still remains areas that are difficult for vision-based navigation.

A more detailed look at feature matches is presented in Figure 5.18. The top figure shows the distribution of total inlier matches across all channels for each traverse, and the bottom figure shows the distribution of inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery. The background of the plots (vertical bars) are colored with respect to daylight conditions. Note the three instances of night driving (dark purple) on the far right.

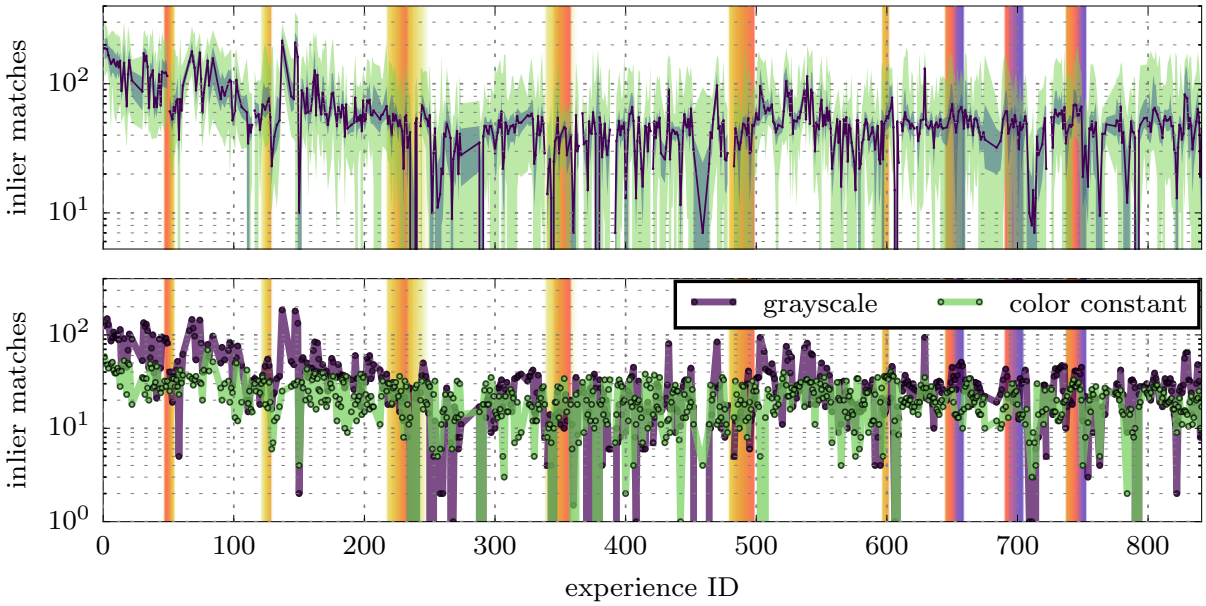


Figure 5.18: Inlier match count for the Ethier Gravel pit field test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel. These results show that for the majority of autonomous traverses in this field test, the inlier match count remained relatively high, with a median value of between 100 and 30 inliers from all channels.

The top figure shows that for the majority of traverses, the median inlier count value (dark purple line) stays between 100 and 30 inlier matches. However, there are instances of median values dropping to single digits with Q1 values (light shaded area) of zero. These cases can be attributed to navigation in environments challenging for vision-based navigation during difficult weather conditions. These include high winds in vegetation-rich areas, sunshine and terrain modification in open desert areas, strong shadows on tree-lined roads, and glare when the elevation of the sun is low. A detailed analysis of these corner cases can be found in Section 5.6. The bottom figure shows the distribution of inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery. It can be seen that color-constant imagery significantly contributes to the inlier count of the algorithm, with inlier counts higher than grayscale for the 4th and 5th day of the field test (experiences 225-500), where the weather conditions were sunny.

Computation Time Results of the field test with respect to localization computation time is reported in Figure 5.19. This figure shows the distribution of localization computation times for each autonomous traverse. the dark-purple line shows the median values of the traverses, the dark-green, shaded area shows the upper (Q2) and lower (Q1) quartile, and the light-green, shaded area shows the min/max inlier values. The background of the plots are colored with respect to daylight conditions. Note the three instances of night driving on the far right. This figure shows that the median localization computation time for most traverses is below or near the 100 ms mark. This value is safely within the tolerance for online driving for the VT&R 2.0 system, whose parallelization allows for VO at the frame rate of the sensor and localization at the rate of keyframe creation which is between 2 and 4 Hz.

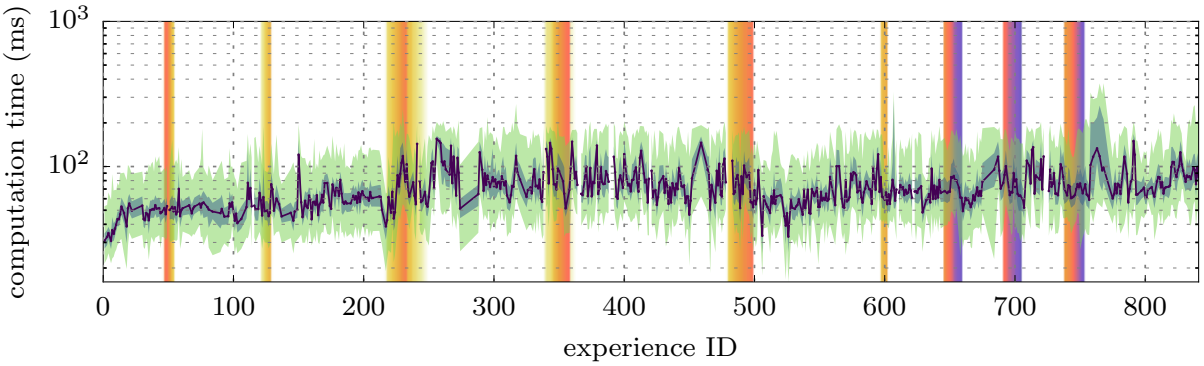


Figure 5.19: Localization computation time for the Ethier Gravel Pit Field Test. These results show that localization for the VT&R 2.0 system typically stayed below the 100 ms mark, which is well below the tolerance of the parallelized state estimation of VT&R 2.0 (see Section 5.2.6)

Detailed Results

The overall results of this field test has shown that in general, the VT&R 2.0 system provided safe, reliable autonomous path following. However, there are a few outlier cases where inlier feature matches were sparse, localization failures were prevalent, and distance on dead reckoning was higher than usual. This is primarily due to specific environments encountered during this field test that are difficult for vision-based navigation.

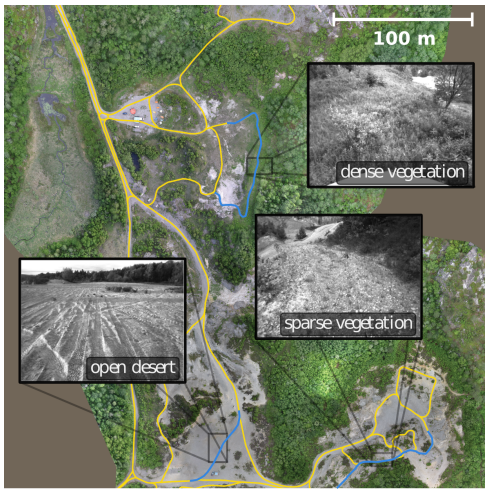


Figure 5.20: Areas of varying difficulty for vision-based navigation encountered during the Ethier field test.

This section aims to demonstrate and quantify the impact that the environment has on the performance of the localization system. To do so, we present detailed results on specific paths in the network that are easy, moderate, and difficult for vision-based navigation. These areas are highlighted in Figure 5.20. An area that is easy for vision-based navigation contains sparse, short vegetation, large rocks, trees, and sand. Detailed results for this area are presented in Section 5.5.1. An area that is moderate for vision-based navigation contains dense vegetation, tall grass, and a tall treeline. Detailed results for this area are presented in Section 5.5.1. An area that is ex-

tremely difficult for vision-based navigation contains an open desert with shifting sand and no vegetation for hundreds of meters. Detailed results for this area are presented in Section 5.5.1.

Figure 5.21: Appearance change in a sparsely vegetated area of the 5 km network of paths. In this section of the network, appearance change manifested primarily from lighting.

Sparse Vegetation This section provides results for a 91 m section of the network whose appearance is shown in the center image of 5.20 and corresponding path is highlighted as a blue line. The path, consisting of sand, gravel, and sparse vegetation, starts at the top of a steep ridge and descends gradually to the bottom of an untended gravel pit. We chose to highlight this section of the network as an area that is particularly easy for the VT&R 2.0 system. This area is not challenging for our system because its tends to only change due to lighting. This can be seen in Figure 5.21, which highlights the appearance change seen during the field test. The appearance change is due mostly to shadows moving across the scene and glare in the image during sunset. Apart from this, the short sparse vegetation does not change much over time, causing stable features to persist over longer periods of time. As a result, this stretch of the network was autonomously traversed 56 times with an autonomy rate of 100%.

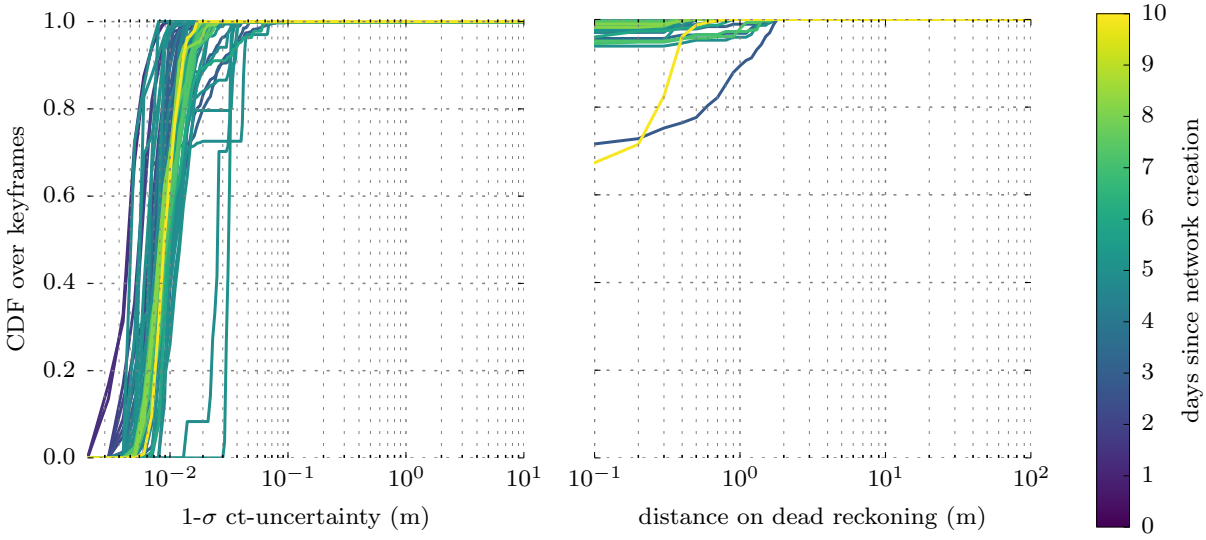


Figure 5.22: Localization Results for all 56 autonomous traverses of the sparse vegetation path in the 5 km network of paths of the Ethier Gravel Pit field test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses. *note:* log scale on x axes.

Results of this stretch of the network with respect to cross-track uncertainty and distance driven on VO are displayed in the left- and right- hand plots of Figure 5.22, respectively. These figures show

that for all runs, the cross-track uncertainty remained below 0.1 m for 100% of each traverse and the distance on VO remained below 2 m. Results of this stretch of the network with respect to feature inlier

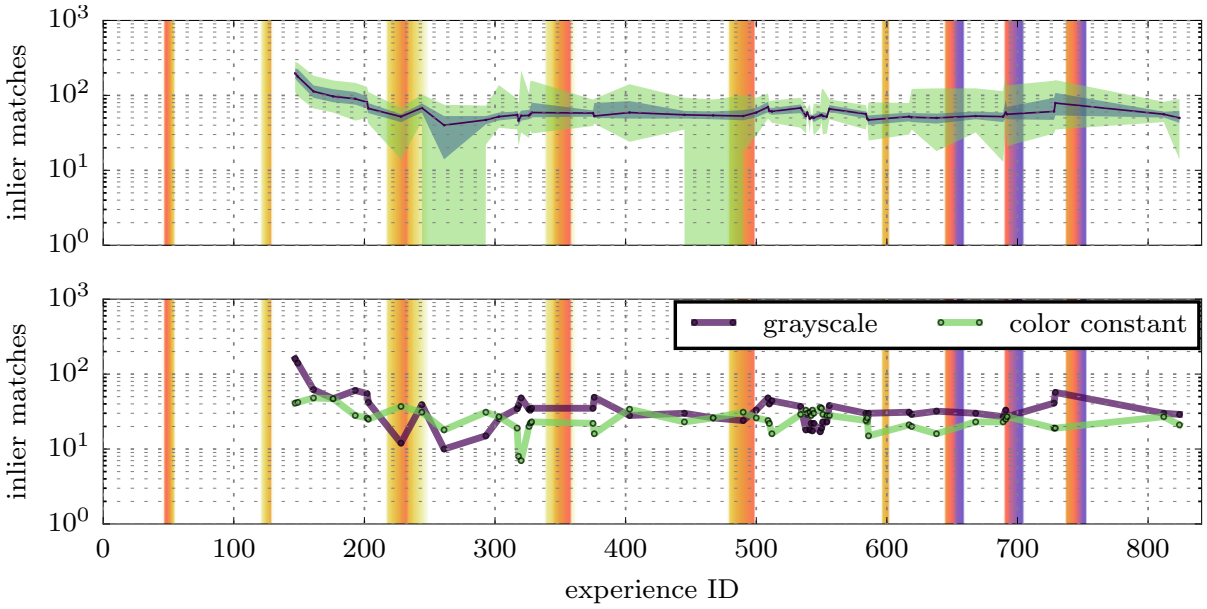


Figure 5.23: Inlier Match count for for all 56 autonomous traverses of the sparse vegetation path in the 5 km network of paths of the Ethier Gravel Pit field test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel.

matches is displayed in Figure 5.23. The top figure shows the distribution of total inlier matches across all channels for each traverse, and the bottom figure shows the distribution of inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery. For all 54 autonomous traverses of this stretch of the network, the median inlier match count stayed above 50 matches, with the upper/lower quartile near the 50 mark as well. The bottom figure shows that both the grayscale and color-constant channels had a near equal impact on total inlier matches.

Dense Vegetation This section provides results for the 240 m section of the network whose appearance is shown in the top image in Figure 5.20 and corresponding path is highlighted as a blue line. The path begins in a meadow with tall grass and rapidly inclines up to a ridge containing thick vegetation bordered by a tree line, finishing with a steep descent into a sandy gravel pit. This area of the network is moderately difficult for vision-based systems due primarily to the combination of tall grass and tree line. This can be seen in Figure 5.21, which highlights the appearance change seen during the field test. The proximity to large trees causes the path to be enveloped in harsh shadows that rapidly move in windy conditions and quickly move throughout the day with the elevation of the sun. Tall grass sways in the wind, confounding geometric consistency checks and grows in height over modest time scales. Despite these challenges, the robot autonomously repeated this route 109 times with only two major manual interventions. Examples of the appearance change seen in this section is displayed in Figure 5.24, note the image on the right which shows an autonomous traverse at night with on-board headlights.

Results of this stretch of the network with respect to distance driven on VO and cross-track uncertainty are displayed in the left- and right- hand plots of Figure 5.25. For the majority of traverses, the cross-track uncertainty remained below 0.2 m, with the exception of a few traverses, where the uncertainty exceeded 5 m for only 1% of the distance traveled. These anomalies can be attributed to VO

Figure 5.24: Appearance change in a densely vegetated area of the 5 km network of paths. In this section of the network appearance change manifested from external lighting, growing vegetation, and vegetation moving swiftly in the wind.

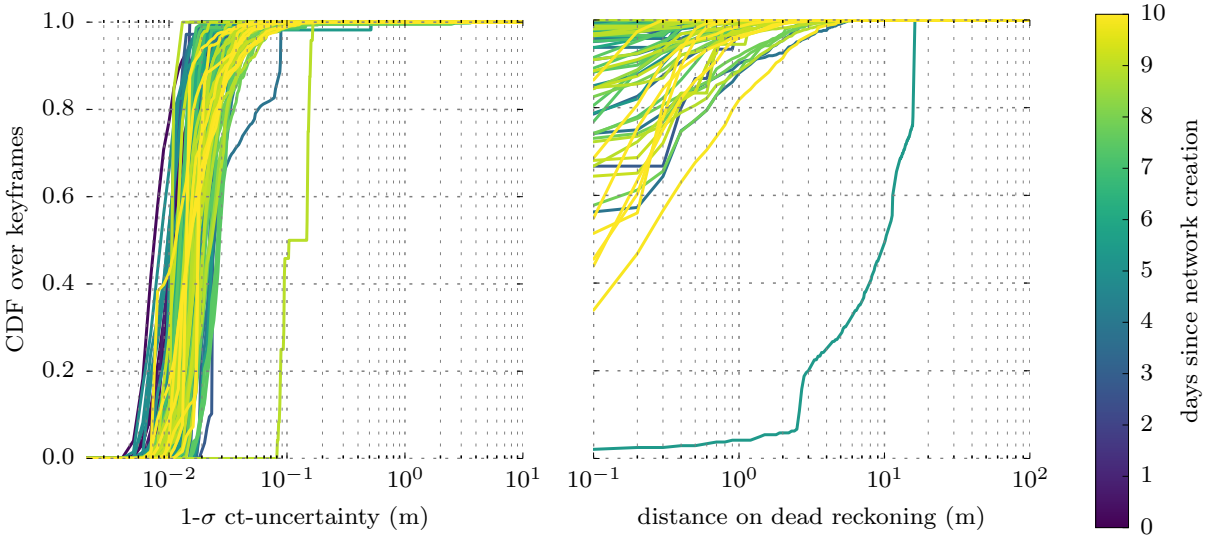


Figure 5.25: Localization Results for all 109 autonomous traverses of the dense vegetation path in the 5 km network of paths of the Ethier Gravel Pit field test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses. *note:* log scale on x axes.

failures during the night-time traversals, which can cause spikes in the uncertainty of the estimator. For all but one of the traverses in this section of the path, the distance driven on dead reckoning remained below 5 m for 100% of each traverse and below 1 m for 80% of each traverse. The one exception, where the distance driven on dead reckoning exceeded 15 m occurred during high winds, causing the vegetation to sway back and forth.

Results of this stretch of the network with respect to inlier feature matches are displayed in Figure 5.26. The top figure shows the distribution of total inlier matches across all channels for each traverse, and the bottom figure shows the distribution of inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery. For the majority of the 109 autonomous traverses of this stretch of this network, the median inlier match count stayed above 30. However, there are a handful of runs where the median count is as low as ten matches, with one run having a median inlier count of zero. These difficult runs occurred during sunrise/sunset when there is glare in the camera, and during times of high wind where the vegetation sways back and forth.

The color-constant images in this section of traverse significantly contributed to the performance of the localizer. There are many runs, especially starting on the fourth day (exp. ID $>$ 350), where the

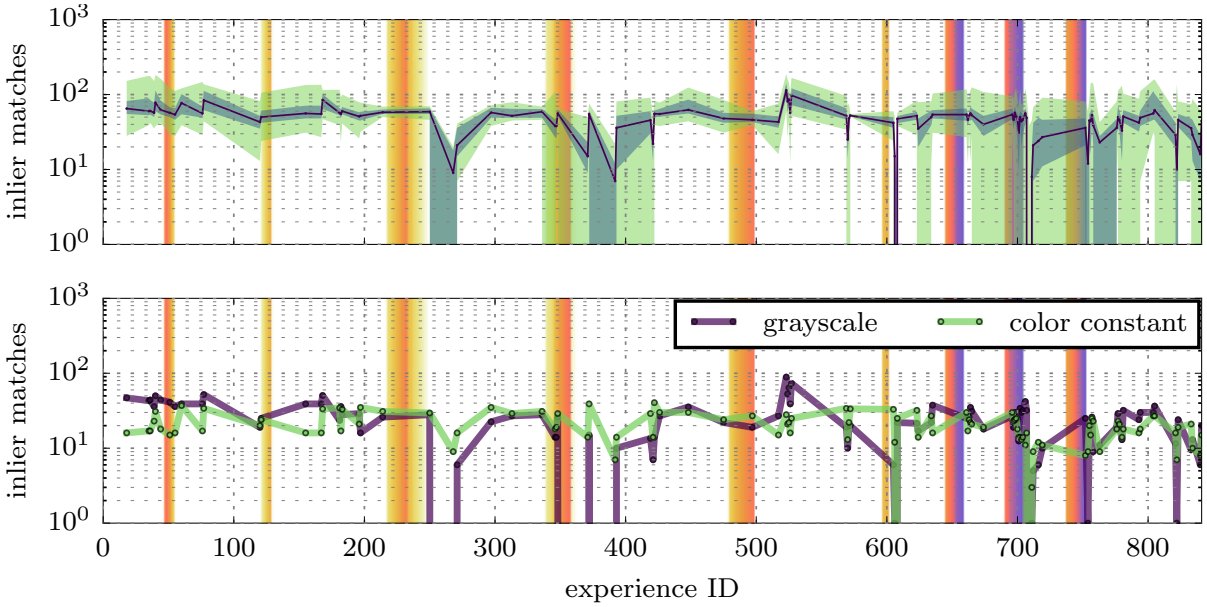


Figure 5.26: Inlier Match count for for all 109 autonomous traverses of the dense vegetation path in the 5 km network of paths of the Ethier Gravel Pit field test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel.

median color-constant count is much higher than its grayscale counterpart.

Open Desert This section provides results for a small area of the network whose appearance is shown in the top image in Figure 5.20 and corresponding path is highlighted as a blue line. The path intersects an open, flat desert area with little to no vegetation. This area is especially challenging for vision-based navigation because there are no stable features in the scene with the exception of the tree line which is on the horizon. This can be seen in Figure 5.27, which highlights the appearance change seen during the field test. Close features on the ground in this area are ephemeral due to the sand shifting in the wind on a daily basis, and vehicles driving through the scene creating tire tracks. Much like the snow-filled meadows discussed in Section 3.6.3, the lack of stable ground features cause inter-day features to be limited to the horizon of the scene. For a stereo-based system, these features will have extremely high depth uncertainties and only contribute to the rotation estimate of the robot.

Results of this stretch of the network with respect to distance driven on VO and cross-track uncer-

Figure 5.27: Appearance change in an open desert area of the 5 km network of paths. In this section of the network, appearance change manifested from external lighting, shifting sand, and vehicle tracks. The lack of stable features apart from the treeline on the horizon make this environment especially difficult for vision-based navigation.

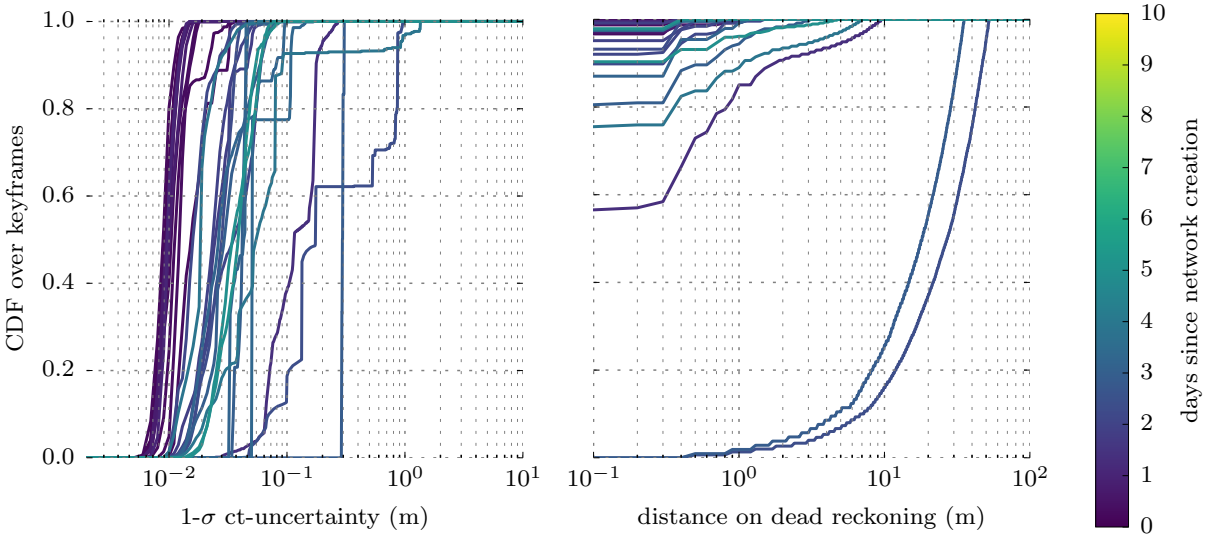


Figure 5.28: Localization Results for all 31 autonomous traverses of the open desert path in the 5 km network of paths of the Ethier Gravel Pit field test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses. *note:* log scale on x axes.

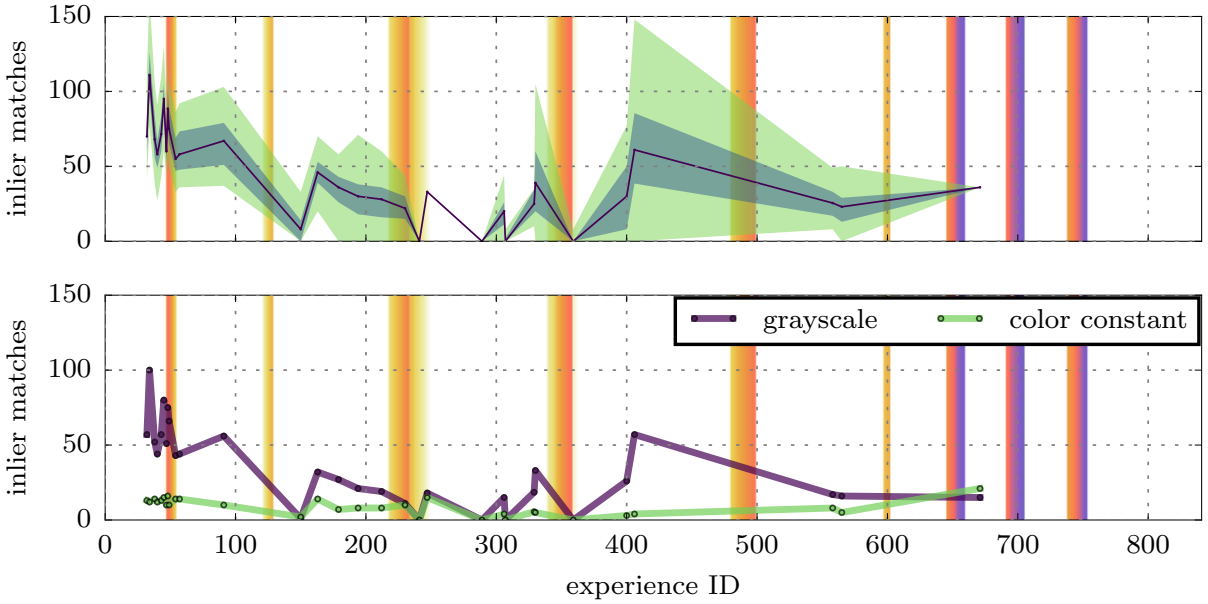


Figure 5.29: Inlier Match count for for all 31 autonomous traverses of the open desert path in the 5 km network of paths of the Ethier Gravel Pit field test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel.

tainty are displayed in the left- and right- hand plots of Figure 5.28. The uncertainty for traverses in this area are overall much larger than previous areas of the network. For the majority of traverses, the 1-σ cross-track uncertainty remained below 0.05 m for only 50% of each traverse and remained below 0.1 m for 100% of each traverse. For five traverses, the uncertainty was much higher with values exceeding 1 m. This increased uncertainty is due in part to stable features being limited to the horizon, and an increase of localization failures causing longer distance on dead reckoning which is much higher than other sections of the network.

This can be seen in the right-hand side of Figure 5.16, which shows a marked increase in distances driven on dead reckoning compared to the sparse and dense vegetation areas. For two traverses of this path, the distance on dead reckoning exceeded 40 and 50 m. These long lines indicate that the robot was on dead reckoning for nearly the entire section of the path. These two traverses caused the robot to be significantly out of its tracks and demanded manual interventions. Results of this stretch of the network with respect to inlier feature matches is shown in Figure 5.29. These plots show that feature matches in this section was low overall compared to other areas of the network, with median values dropping to zero on three traverses. It can also be seen that the color-constant images provide little support in this area of the network. On the eighth day of of testing, after too many major manual interventions, this small stretch of the network was deemed untraversable by the VT&R 2.0 system and abandoned.

Discussion

Difficult Conditions For the majority of the 140 km of autonomous driving, the median inlier match count was at or above a safe value of 50, as shown in Figure 5.17. However, there were a select few traverses with median values near 10 matches and Q1 values of zero. This poor localization performance can be attributed to traversing in conditions that are difficult for vision-based navigation. These difficult conditions that were encountered during the field test are highlighted in Figure 5.30. Open, desert areas contain few features that persist over time with vehicles, wind, and weather changing the shape of the sand on a daily basis, which causes any stable features to be limited to the horizon. Vegetation-rich areas are typically not a problem for the VT&R 2.0 system, except when high winds are present, which causes the vegetation to rapidly sway back and forth, causing issues for outlier rejection for both localization and VO. Tree-lined corridors cast strong shadows on the road on sunny days. In these conditions, the

Figure 5.30: Example images of areas that remain difficult for vision-based navigation. from left to right: i) open desert areas with heavy vehicle traffic, ii) lush vegetation during high winds, iii) tree-lined corridors with strong shadows, and iv) sun glare.

majority of inlier matches originate from these ephemeral features. With multi-experience systems, this can lead to incorrect state estimates if the majority of inlier matches arise from features that have all moved with the elevation of the sun. The final difficult condition encountered during the field trial is glare from when the sun is low on the horizon, causing images to be oversaturated. It was in these conditions on the sunniest days of the field test where the majority of manual interventions occurred.

Conclusion

This field test demonstrated the VT&R 2.0 system's ability to provide large-scale, autonomous path following over intra-seasonal appearance change. During this field test, a 5 km network of connected paths was demonstrated to the robot in a challenging, unstructured, outdoor environment. Over an eleven day period, the robot continuously traversed the network, accumulating over 140 km of autonomous driving, with an autonomy rate of 99.59% of distance traveled. While the major manual interventions that occurred during the field test were the result of environments that remain challenging for vision-based navigation, many of the minor and all of the trivial manual interventions were the result of implementation issues in the VT&R 2.0 system that were present during this field test. Known software bugs required frequent trivial manual interventions to continue operation (see Section 5.5.1). Unknown software bugs at the time of the field test caused slow downs in the VO pipeline and sporadic errors in the VO estimate, which can adversely affect path tracking and localization. In the field tests presented hereafter, these bugs have been addressed and an increase in system performance is apparent in the results presented in the upcoming sections.

5.5.2 UTIAS In The Dark

This section presents results on the UTIAS In The Dark Field test (Section 5.3.3). These results show that the VT&R 2.0 system, through the use of the MEL algorithm, is capable of operation across a full diurnal cycle, with on-board headlights at night. We show both stable localization across this extreme appearance change and online performance despite the map containing over 40 experiences.

Cross-Track Uncertainty

Results of the field test with respect to cross-track uncertainty are displayed in Figure 5.31. The left-hand figure shows the CDF of the $1-\sigma$ cross-track uncertainty for all 40 autonomous traverses of the field test. These results show that for all autonomous traverses the distribution of cross-track uncertainty remained the same, staying below 0.1 m for 100% of each traverse, and below 0.02 m for 50% of each traverse. Traverses that occurred during the day (dark purple and bright yellow), have slightly better uncertainties, remaining below 0.02 m for 100% of the traverses, while traverses at night and during sunrise/sunset (light blue, green) have slightly higher uncertainties. This can be explained by the glare in the images during sunrise/sunset (see Figure 5.10) and the smaller amount of visual features observed at night due to the scene being only partially illuminated by on-board headlights. We refer the readers to MacTavish and Barfoot (2017) for a detailed analysis on the performance of the VT&R 2.0 VO system at night with on-board headlights.

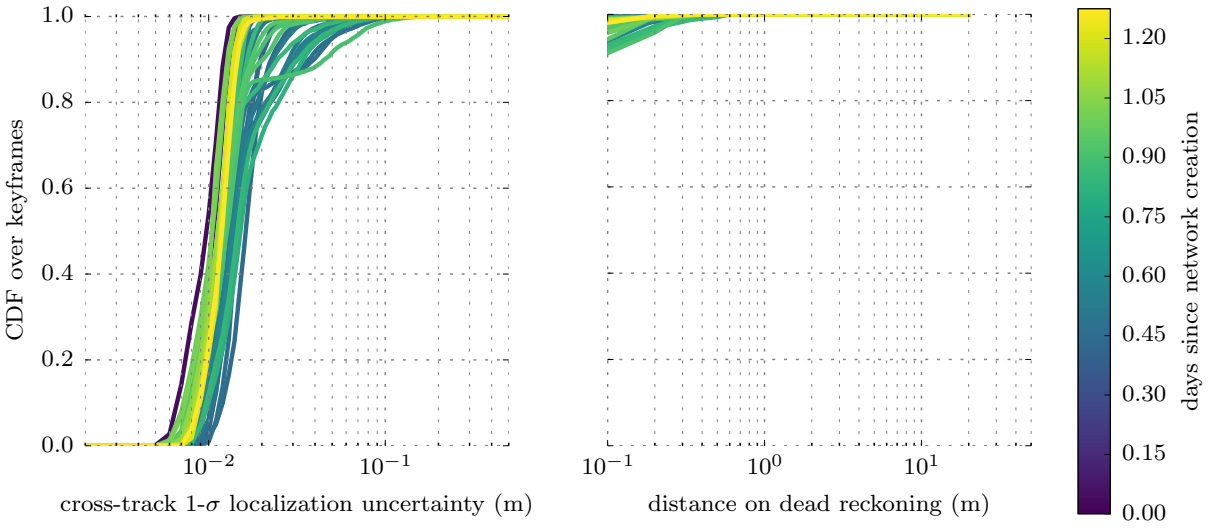


Figure 5.31: Localization Results for all 40 autonomous traverses of the UTIAS in the dark test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses. These results show that for all 40 autonomous traverses, the cross-track uncertainty remained below 0.1 m and the distance on dead reckoning remained below 0.5 m.

Distance on Dead Reckoning

Results of the field test with respect to distance driven on dead reckoning are displayed in the right-hand side of Figure 5.31. These results corroborate the low cross-track uncertainty observed during this field test, with all autonomous traverses driving less than 0.5 m on dead reckoning for 100% of each traverse. Again, traverses driven at night (green lines) show slightly higher distances on dead reckoning.

The VT&R 2.0 system only localizes at the rate of keyframe creation, which is roughly every 0.3 m for these field tests. Distances driven on dead reckoning for these small amounts are likely due to isolated localization failures and isolated localization solves that are slower than the rate of keyframe creation.

Feature inlier count

Results of the field test with respect to inlier feature matches are shown in Figure 5.32. The top figure shows the distribution of total inlier matches across all channels for each traverse. For all autonomous traverses in this field test the median inlier feature count (dark, purple line) over the entire traverse remained above 50 features despite extreme changes to lighting conditions. The upper and lower quartile (dark shaded area) remained above 40 inlier matches and the min/max (light, shaded area) stayed above 25 inliers for the majority of traverses.

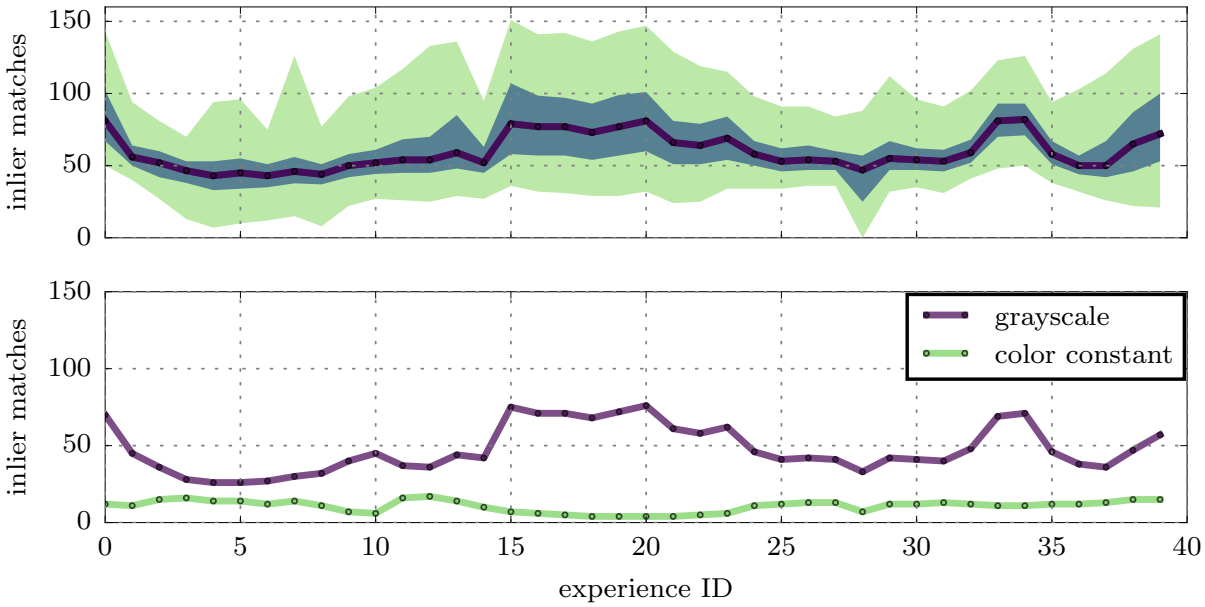


Figure 5.32: Inlier Match count for the UTIAS In The Dark Field Test. *top:* Total inlier match distribution for all information channels. *bottom:* Inlier match distribution for each information channel. These results show that for all autonomous traverses in this field test, the inlier match count remained relatively high, with a median value of 50 inliers from all channels.

The bottom figure shows the distribution inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery. For this field test, the color-constant imagery did not contribute significantly to the performance of the localizer. This is due primarily to the frequency of experiences collected in this field test with respect to the appearance change. Because an experience is gathered roughly every hour, there is always a large amount of grayscale matches to the previous experiences.

Computation time

Results from the field test with respect to localization computation time is shown in Figure 5.33. The results show that despite accumulating over 40 experiences in the map at the final run, the median computation time and upper quartile remain below 100ms. This steady computation time is because in

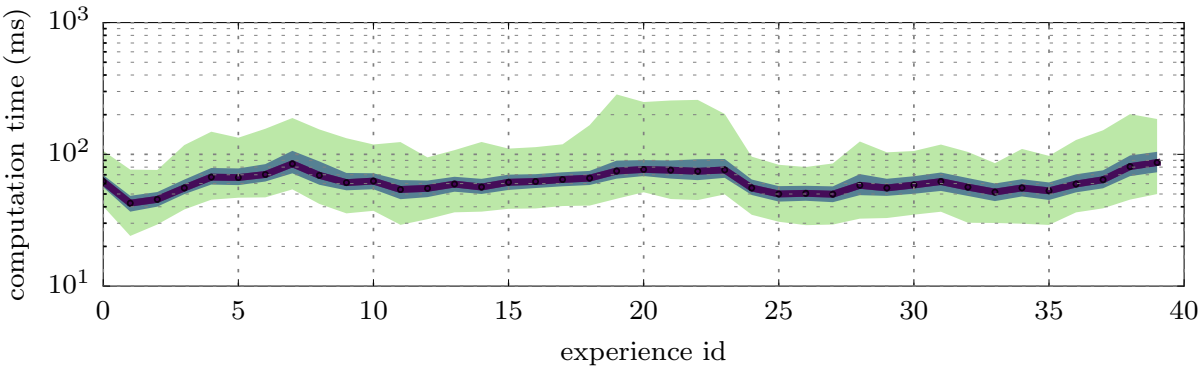


Figure 5.33: Localization computation time for the UTIAS in the dark field test.

this field test, the BoW experience selector (Section 5.2.5) is used to pick the 10 experiences in the map that are most similar to the live view to be used in the MEL optimization problem. There is a noticeable increase in the maximum computation time between runs 17 and 22, with values reaching 200-300 ms. This are runs that occurred at night with on-board headlights and are likely due to failures in VO, which cause an increase in uncertainty and an increased localization window, which expands the number of vertices the MEL system localizes against for the next frame, temporarily increasing the computational cost of localization.

Conclusions

This section presented the results of the performance of the full vision-in-the-loop VT&R 2.0 system with respect to the UTIAS In The Dark field test detailed in Section 5.3.3. The primary objective of this field test was to demonstrate the VT&R 2.0 system's capability to perform vision-in-the-loop autonomous path following across extreme appearance change over a full diurnal cycle outdoors. This field test furthermore demonstrates the system's ability to computationally bound the MEL algorithm using the BoW experience selector. These results have validated this claim, showing the system's ability to provide robust, accurate localization between the privileged view and all autonomous traversals. During this field test the robot autonomously traversed over 18 km with an autonomy rate of 100%, a median inlier count above 50, and a maximum cross-track uncertainty below 0.5 m, despite appearance change as extreme as night vs. day. The next section demonstrates the system's ability to perform long-term autonomous path following, over extreme seasonal appearance change.

5.5.3 UTIAS Multi-Season

The results of the previous field tests have demonstrated the VT&R 2.0 system's ability to provide large-scale, autonomous path-tracking across appearance change due to lighting and weather over modest time scales. This section presents results on the UTIAS multi-season field test detailed in Section 5.3.4. These results show that the VT&R 2.0 system is capable of long-term autonomy over multiple seasons. We demonstrate reliable autonomous path following over a 100 day period from winter to summer, where the appearance of the scene transitions from deep snow and freezing rain to lush summer vegetation. During the field test, the robot autonomously traversed a 165 m loop in this environment 163 times accumulating over 27 km of autonomous driving and maintaining an autonomy rate of 99.98%.

Autonomy Rate

During this multi-season field test, the robot autonomously repeated the 165 m loop fully 163 times and partially 6 times between the dates of 01/31 and 05/27 with an autonomy rate of 99.98% of distance traveled. Occasionally, small manual interventions were required to maneuver the robot across inclines during conditions of deep snow and slick mud.

Partial traverses were cut short when the VT&R 2.0 system could no longer continue the path. For all but one of these attempted traverses, the failure point was the XB3 camera sensor. During the winter months, four attempts at traversing the loop were cut short due to an excessive amount of glare in the lens caused from bright sunshine reflecting off deep snow. On one rainy day in the spring, the combination of humidity and cold caused the camera lens to fog up. In all five of these cases, an increase in uncertainty due to VO and localization failures triggered the VT&R 2.0 system to stop the robot in its tracks. For each attempt, the experiment was either concluded and the robot was manually driven back to the starting point, or the VT&R 2.0 system was commanded to autonomously traverse back to the starting point. The final partial traverse of the loop was cut short due to the robot being noticeably out of its tracks. At this point, the VT&R 2.0 system was commanded to repeat back to a safe position, and the traverse was successfully completed. The two partially attempted traverses where the robot autonomously repeated back to its starting location demonstrates the system's ability to autonomously recover from failure conditions. The field test was concluded on 05/23 when localization consistently began to fail in the tall grass after 6 days of no autonomous traversals. This is due to the lack of autonomous operation during this time of rapid appearance change. A discussion on this failure point is reserved for Section 5.5.3.

Cross-Track Uncertainty

Results of the field test with respect to cross-track uncertainty are displayed in Figure 5.34. This figure shows the CDF of the $1-\sigma$ cross-track uncertainty for all 163 autonomous traverses. The results show that for most traverses, the uncertainty remained below 0.2 m, with the exception of 5 traverses, where the uncertainty rose to 1 m. It is also interesting to note that the uncertainty increases as the delta time between the traverse and the taught path approaches 13 days (dark to light purple) and then decreases again as the delta time approaches 100 days (light purple to bright yellow). This is due to an increase in uncertainty during days with deep snow and bright sunshine. After the deep snowfall, the only common landmarks to previous bridging experiences are on the horizon.

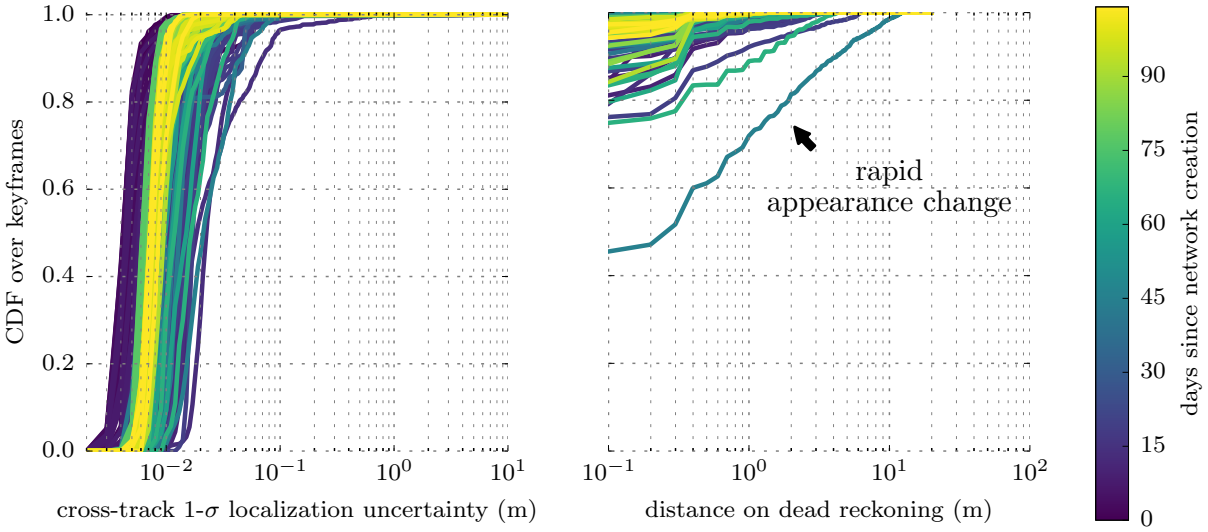


Figure 5.34: Localization Results for all 163 autonomous traverses of the UTIAS multi-season field test. *left:* CDF of the $1-\sigma$ cross-track uncertainty for all traverses. *right:* CDF of the distance driven on dead reckoning for all traverses. These results show that for all 178 autonomous traverses, the cross-track uncertainty remained below 1 m and the distance on dead reckoning remained below 15 m.

Distance on Dead Reckoning

Results of the field test with respect to distance driven on dead reckoning are displayed in the right-hand side of Figure 5.34. This figure shows the CDF of the distance driven on dead reckoning for all 163 autonomous traverses. The results show that for the majority of traverses, the distance driven on dead reckoning remained below three meters. For three traverses, the distance driven on dead reckoning exceeded 3, 6, and 12 meters. These traverses are notable times of extreme appearance change, occurring directly after snow fall or melt. These results show that traverses with high distances driven on dead reckoning tend to correspond to runs with higher than usual uncertainty.

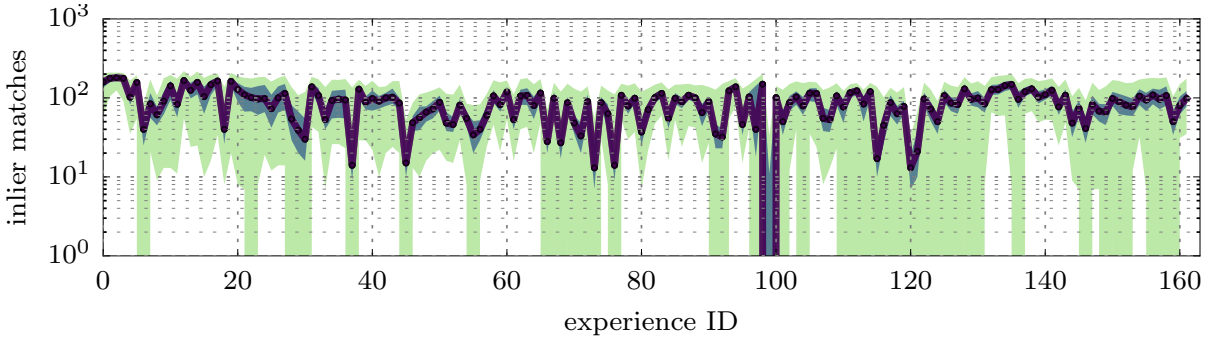


Figure 5.35: Inlier match count for the UTIAS multi-season field test. This figure shows the total inlier match distribution for all information channels.

Feature inlier count

Results of the field test with respect to inlier feature matches are shown in Figure 5.35 and Figure 5.36. Figure 5.35 shows the distribution of total inlier matches across all channels for each traverse. These

results show that for the majority of traverses, the median inlier match count (purple line) fluctuated between 20 and 150 inliers, with the upper and lower quartile values remaining close to the median. However, there were a handful (< 10) traverses where the median count dropped below 20 inlier matches, with one example of a median match count of zero. These difficult traverses are due to rapid appearance change due to snow fall and snow melt in the winter months.

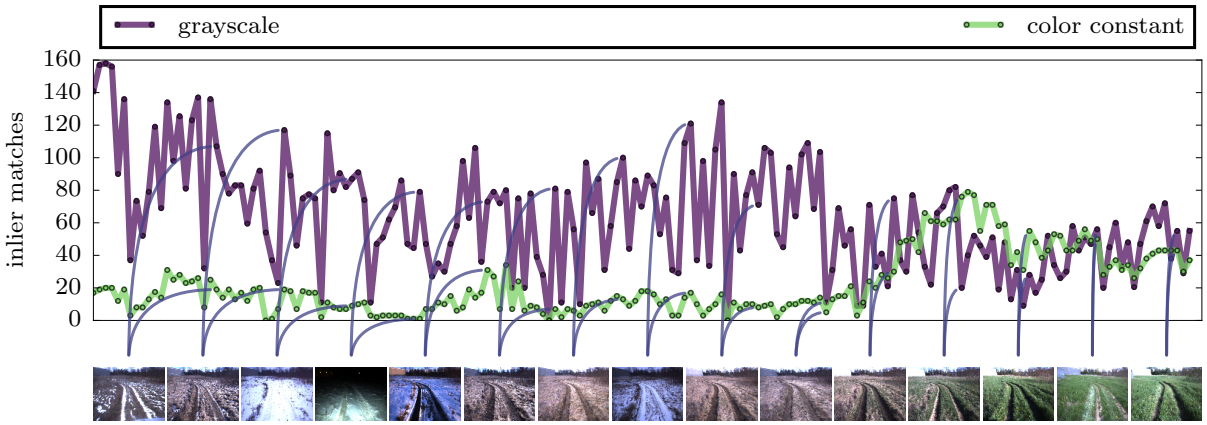


Figure 5.36: Impact of the color-constant images during the 2017 UTIAS multi-season field test.

Figure 5.36 shows the distribution of inlier matches between the two information channels used in this field test: grayscale and color-constant stereo imagery as well examples of the appearance of the scene. This figure validates the hypothesis formulated in Chapter 3: that the performance of localization with color-constant images is highly dependent on the environment. For all of the VT&R 2.0 field tests, the *Forest CC* color-constant image transformation was used. This image was experimentally tuned to perform well in green, vegetation environments. We furthermore have shown in Chapter 3 that localization performance with color-constant images in winter environments is generally poor. This figure shows that inlier feature matches from the color-constant images had little impact on the field test until the grass began to grow in the spring. It can be seen that once the appearance of the scene transitioned from dead vegetation to green grass the impact of the color-constant images dramatically increased, outperforming the inlier match count from grayscale images for a period of time.

Computation time

Results from the field test with respect to localization computation time and number of experiences used in localization is shown in Figure 5.37. The top figure shows the distribution of localization computation times and the bottom figure shows the number of experiences the selection algorithm recommended for use. As discussed in Section 5.3.4 and detailed in Table 5.4, the algorithm that selects which experiences to use localization varied throughout the field test. At experience 110, the selection algorithm was changed from Time of Day (ToD) selection to Collaborative Filtering (CF). This change can be seen in the bottom figure when the number of experiences recommended between traverses begins to vary more.

For the first 38 autonomous traverses, the number of experiences used varied between 5 and 10, During this time, the median computation time remained steady at 50 ms with the maximum values falling between 100 ms and 200 ms. On the 39th experience, the number of recommended experiences was increased to accommodate difficult localization conditions in the deep snow, and remained at this level until experience 120. For these autonomous traverses, the MEL computation time fluctuated

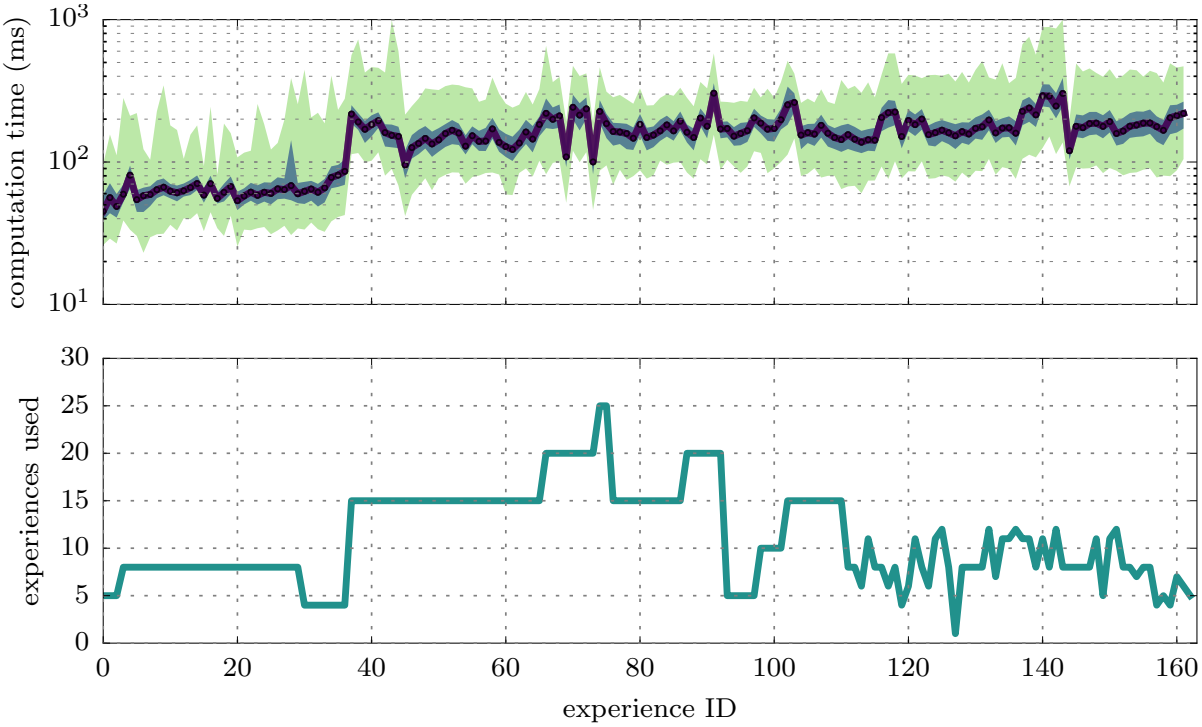


Figure 5.37: Localization computation time for the UTIAS multi-season field test. *top:* Median localization computation time for all 163 autonomous traverses of the UTIAS multi-season field test. *bottom:* The number of experiences used in the MEL localization state estimate.

between 100 and 200 ms with maximum values typically below 400 ms, but sometimes reaching 700 ms. At experience 110, the experience selection algorithm changed to CF, which recommends between 5 and 10 experiences. Because this algorithm is more computationally expensive compared to the ToD experience selector, the localization computation time does not change despite the drop in total amount of experiences recommended.

Discussion

This section discusses lessons learned and key insights into the performance of the VT&R 2.0 system while performing this multi-season field test.

Appearance-change Rate vs. Operational Frequency This field test demonstrated the VT&R 2.0 system's ability to perform autonomous path following over seasonal appearance change as extreme as winter vs. summer. However, in order to be successful, the robot is required to periodically traverse the path to obtain a sufficient amount of experiences in order to bridge the appearance gap between the live and privileged experience. This requires a demand on the application to keep to a schedule of operation that sufficiently bridges the appearance gap to guarantee success. This schedule was provided to the rover by a human operator during the field tests presented in this thesis.

When lighting change is the only factor, it is sufficient for the algorithm to gather only a modest amount of experiences to capture the different lighting conditions seen throughout a diurnal cycle. When weather and seasonal changes are a factor, the schedule the robot needs to keep to guarantee a sufficient amount of bridging experiences is highly variable. In the snowy fields of the UTIAS campus,

the appearance can dramatically change over a period of hours due to snow fall and snow melt. In order to ensure reliable autonomy in this environment, the robot would be required to constantly drive during or shortly before and after snow storms. In the spring, the appearance changes much slower and a repeat every few days is sufficient. However, in the summer, the grass in the field rapidly grows to a height of nearly 1.5 m, demanding a more regular autonomy schedule. It was during this time of rapid vegetation growth when autonomy failed and the filed test came to a conclusion. This failure point was due to a lack of operation over a period of six days as well as the difficult localization conditions of tall vegetation. This requirement of constant operation to address rapid appearance change is a potential issue for many commercial operations and more work to automatically generate autonomy schedules and mitigate the demand for constant autonomy is merited.

Overconfident bridging experiences In the Photocopy of a Photocopy (PoaP) field test (Section 4.4.3), we demonstrated the VT&R 2.0 system's ability to indirectly localize to a privileged, manually driven experience through bridging experiences who themselves have increasingly inaccurate localizations to the privileged experience. While spatial drift was evident, it was manageable and had little effect on the system's ability to accurately repeat the manually taught path.

The success of the PoaP field test was in part due to the short temporal distance between each bridging experience, allowing the localizer to maintain excellent performance over the duration of the test (see Section 4.6.3). However, spatial drift increases significantly faster in situations where there are an insufficient amount of experiences in the map to bridge the appearance gap between the live and privileged experience. In scenarios where localization estimates to the privileged experiences are less accurate than usual, it is vital to the health of the system to report proper uncertainties. If the localizer is failing during an autonomous traverse, the robot may drift out of its tracks, and remain out of its tracks until it is corrected by an accurate localization. If overconfident, inaccurate localizations are stored in a bridging experience that has captured the new appearance of the scene, then the robot will become "locked in" to the new experience and will remain out of the original tracks on subsequent traverses. If this process occurs too often, then the autonomous traversal of the path may eventually fail. This spatial drift due to overconfident estimates was observed during the UTIAS multi-season field test. During this field test, there were instances of autonomous traverses where localization performance was overall very poor. This was typically after sudden appearance change due to heavy snow fall or snow melt. During these traverses, the robot veered out of its tracks in isolated areas of the path by up to 30 cm. During this inaccurate path tracking, the uncertainty of the estimator increased, but not at the rate at which the robot was out of its tracks. This caused the robot to become locked into the new experience for either the remainder of the field test or until a "temporal loop closure" was satisfied.

This issue slowly manifested itself over the 104 day field test. While the drift was never large enough to abort the path due to unsafe driving, it is a serious issue that needs to be resolved in order to perform true long-term autonomy. A discussion on how to resolve this issue is presented in Section 5.6.

Conclusions

This field test demonstrated the VT&R 2.0 system's ability to provide long-term, autonomous path following over inter-seasonal appearance change. The field test consisted of autonomously traversing a 165 m loop demonstrated to the robot in a meadow on the UTIAS campus. Over a 104 day period, the robot periodically traversed the network 163 times, accumulating over 27 km of autonomous driving,

with an autonomy rate of 99.98% of distance traveled. Over the duration of the field test, the appearance of the scene transitioned from deep snow and freezing rain in the winter to lush, green vegetation in the summer. The primary objective of this field test was to demonstrate the system's ability to operate across long-term seasonal appearance change. This objective was met, with overall stable localization performance results across extreme appearance change. However, during the field test we exposed a few key system issues, discussed in Section 5.5.3, that need to be addressed in order for it to be viable for real-world use. We provide a discussion on potential solutions to address these issues in Section 5.6.

5.6 Discussion

The results presented in the previous section demonstrated the VT&R 2.0 system's capability to provide large-scale, long-term, vision-based autonomous path following in unstructured outdoor environments. In Section 5.5.1, we demonstrated large-scale, autonomous path following with the VT&R 2.0 system across intra-seasonal appearance change due to lighting and weather. In Section 5.5.2, we demonstrated autonomous path following across a full diurnal cycle. In Section 5.5.3, we demonstrated long-term autonomous path following across inter-seasonal appearance change.

In total, these results covered over 178 km of vision-in-the-loop, autonomous path following with an autonomy rate of 99.7% of distance traveled. The small amount of manual interventions required during field testing of the VT&R 2.0 system were typically due to localization failures in conditions and environments that are difficult for vision-based navigation. Examples of these difficult conditions are open desert areas with little visual features to associate between experiences, areas containing tall vegetation that sways back and forth in the wind, and deep snow and sunshine that oversaturate camera images. Potential hardware solutions to these problems include a vision sensor with a higher frame rate coupled with a speeded up VO pipeline to properly handle motion estimates in the face of moving vegetation, and smarter exposure techniques to overcome oversaturation during snow glare.

VO Uncertainty Scaling During the UTIAS multi-season field test, we discovered that estimation in the VT&R 2.0 system is overconfident, causing the robot to "lock in" to previous bridging experiences with sub par path tracking in the rare cases when localization was poor. The overconfidence observed in the MEL localization estimate stems from two places: i) overconfident VO estimates stored as temporal edges in the graph, and ii) map landmarks transformed from a single landmark being treated as independent in (4.15) of the MEL estimate. In order to properly tune the uncertainties in the VT&R 2.0 system, these uncertainties must reflect reality. Tuning of the uncertainties in the VO system can be achieved by scaling the uncertainty in the SURF keypoint position estimates. The impact of the keypoint scaling on the total net uncertainty in the VO estimator can be validated through Normalized Innovation Squared (NIS) and Normalized Estimation Error Squared (NEES) validity tests on preexisting data sets and future field tests with ground truth.

Map Landmark Scaling The MEL algorithm provides long-term localization by transforming landmark matches from many experiences into the coordinate frame of a vertex in the privileged, manually driven experience. During this process, (4.15) accounts for map landmarks being transformed into the privileged vertex, V_d, through uncertain transforms in the map. This process assumes that all landmarks observed in a given vertex are independent from each other.

In reality, landmarks originating from the same vertex are correlated with one another. To account for this correlation in future iterations of the system, we can employ a simple heuristic: given a landmark j, originating from the coordinate frame of vertex, V_b, the revised uncertainty equation is defined as follows:

$$\mathbf{R}_j = \mathbf{Y}_j + n_b^{\alpha}\mathbf{Z}_j, \tag{5.3}$$

where n_b is the number of cost terms transformed from V_b, and α is a user-defined scaling parameter between 0 and 1. When the α parameter is set to 0 the equation is identical to (4.15), when the α parameter is set to 1, the equation weights each landmark's uncertainty by the number of measurements matched in that vertex.

Static Viewpoint Reliance This chapter demonstrated that the VT&R 2.0 system is capable of performing long-term, autonomous path following across extreme appearance change. However, the system's performance was analyzed only on field tests and datasets where the path tracking controller kept the robot accurately on the path. In this ideal situation, the localizer benefits from nearly exact viewpoints between the live view and the multi-experience map. Without further analysis, it is not clear how robust the VT&R 2.0 system is to changes in viewpoint while repeating a path. If the system is reliant on a static viewpoint to localize, then it could have the consequence of confining the rover to the taught path. This would be problematic if the system is equipped with a local planner that would otherwise be capable of avoiding obstacles and exploring areas slightly off path.

5.7 Summary and Novel Contributions

This chapter presented the long-term autonomous path-following system, Visual Teach & Repeat (VT&R) 2.0. By using the novel Multi-Experience Localization (MEL) algorithm presented in Chapter 4, this system is capable of inter-seasonal autonomy using only a single stereo camera. We experimentally validate our work through rigorous field tests totaling over 178 km of vision-in-the-loop autonomous driving, experiencing extreme seasonal appearance change. While we have successfully demonstrated long-term autonomous path following with the VT&R 2.0 system, with overwhelmingly superior performance compared to previous vision-based autonomous path-following systems, there are critical performance issues discussed in Section 5.6 that will need to be addressed in future work in order for the system to be usable by commercial applications. The full VT&R 2.0 system with results from the Ethier field test has been accepted and will appear in the 2017 proceedings of the International conference on Field and Service Robotics (FSR) (Paton et al., 2017a).

In summary, the novel contributions of this chapter are:

1. A vision-in-the-loop autonomous path-following system that makes use of a multi-experience localization and mapping framework to provide inter-seasonal autonomy and nighttime autonomy with on-board headlights.

2. Extensive long-term field tests of the system involving autonomy in unstructured, outdoor environments with rapidly changing appearance, covering over 178 km of vision-in-the loop autonomous driving.

Chapter 6

Summary and Future Work

In this chapter, we provide a summary of the novel contributions presented in this thesis and the publications that arose from them. We also discuss future work to further improve vision-based autonomous path following.

6.1 Summary of Contributions and Publications

The motivation behind this thesis stemmed from the desire to perform long-term, vision-based autonomous path following in unstructured outdoor environments. Vision-based autonomous path following has the potential to enable many exciting industrial applications using inexpensive, commercial sensors. The objective of this thesis was to extend the performance of vision-based autonomous path following, whose operational window was limited to a few hours in outdoor environments where the appearance of the scene quickly changes due to lighting (see Chapter 2). We presented here a practical approach to extending the performance of vision-based autonomous path following through experimentation and rigorous field tests. A common theme throughout this thesis is the adaptation of localization and state estimation ideas and techniques to work practically in the field and support vision-in-the-loop path following. Over the course of eight field tests and more than 200 km of vision-in-the-loop autonomy, we have gained insights into vision-based localization and field robotics and the difficulties of operating in real-world environments and have made several publications and what we hope are useful contributions towards long-term, autonomous path following.

The methods and experiments presented in Chapter 3, where we proposed a novel multi-channel localization framework and two examples involving color-constant images and multiple stereo cameras have appeared in the proceedings of three, full-paper refereed conferences:

- Paton, M., McTavish, K., Ostafew, C., and Barfoot, T. (2015a). It's not easy seeing green: Lighting-resistant visual teach & repeat using color-constant images. In *Proc of the Int. Conf. Robotics and Automation (ICRA)*

- Paton, M., Pomerleau, F., and Barfoot, T. (2015b). Eyes in the back of your head: Robust visual teach & repeat using multiple stereo cameras. In *Proc. of the Conf. on Computer and Robot Vision (CRV)*

- Paton, M., Pomerleau, F., and Barfoot, T. (2015c). In the dead of winter: Challenging vision-based path following in extreme conditions. In *Proc. of Field and Service Robotics (FSR)*

These publications were combined with a formulation of the generic multi-channel localization framework in the journal article:

- Paton, M., Pomerleau, F., MacTavish, K., Ostafew, C. J., and Barfoot, T. D. (2017b). Expanding the limits of vision-based localization for long-term route-following autonomy. *Journal of Field Robotics*, 34(1):98–122

In summary, the novel contributions of Chapter 3 are:

1. A multi-channel localization framework that performs independent tracking of point-based visual features for multiple information channels and fuses data correspondences from all channels into a single state estimation problem.

2. A lighting resistant localization system that uses the multi-channel framework to fuse data correspondences from grayscale images and color-constant images to improve performance across lighting change.

3. A multi-stereo localization system that uses the multi-channel framework to fuse data correspondences from multiple stereo cameras to increase the field-of-view of the localization system and improve performance across general appearance change.

4. An in depth analysis of expected localization performance in varying seasons with insight on the limitations of single-experience localization systems that rely on point-based visual features in difficult winter environments.

5. A methodology to experimentally tune the color-constant image transformations to improve performance in a given environment with respect to visual features tracked across lighting change.

The methods and experiments presented in Chapter 4, where we proposed MEL, a novel multi-experience localization system has appeared in one full-paper refereed conference:

- Paton, M., MacTavish, K., Warren, M., and Barfoot, T. (2016). Bridging the appearance gap: Multi-experience localization for long-term visual teach & repeat. In *IROS*

In summary, the novel contributions of Chapter 4 are:

1. A data structure that relates multiple experiences together metrically.

2. A methodology to metrically localize a live experience to a privileged, manually driven experience using several intermediate experiences gathered during autonomous operation.

3. A methodology to bookkeep uncertainties in the multi-experience localizer, accounting for uncertain map landmarks being used from multiple experiences.

4. Experimental evaluations of the MEL system to validate the core ideas of metric localization using many experiences.

The methods and experiments in Chapter 5, where we proposed VT&R 2.0, a novel multi-experience autonomous path-following system that provides long-term vision-based autonomy in unstructured outdoor environments has been accepted and will appear in the following full-paper refereed conference:

- Paton, M., MacTavish, K., Berczi, L., van Es, K., and Barfoot, T. (2017a). I can see for miles and miles: An extended field test of visual teach & repeat 2.0. In *Proc. of Field and Service Robotics (FSR), to appear*

In summary, the novel contributions of Chapter 5 are:

1. A vision-in-the-loop autonomous path-following system that makes use of a multi-experience localization and mapping framework to provide inter-seasonal autonomy and nighttime autonomy with on-board headlights.

2. Extensive long-term field tests of the system involving autonomy in unstructured, outdoor environments with rapidly changing appearance, covering over 178 km of vision-in-the loop autonomous driving.

6.2 Future Work

In this section we discuss potential future work to further improve the performance of large-scale, long-term autonomous path following.

In Chapter 5, we presented Visual Teach & Repeat (VT&R) 2.0, a large-scale autonomous path-following system that uses the Multi-Experience Localization (MEL) framework to achieve long-term autonomy. Through a rigorous set of field tests, the system's ability to perform vision-in-the-loop autonomy across extreme, multi-seasonal appearance change in challenging outdoor environments was demonstrated. While the field test provided promising results, there are key limitations, as discussed in Section 5.6, that can be improved upon to increase the system's robustness to appearance change.

After reviewing the results of the field test, it was determined the state estimation in the VT&R 2.0 system is overconfident in its VO and localization estimates. This can lead to the robot "locking in" to autonomous traverses that experienced drift from the path with overconfident localization estimates. To overcome this issue, future work needs to be done to properly tune the VO and localization estimators to have uncertainties that reflect reality. This process will involve scaling the uncertainty in visual feature keypoints to pass consistency checks in the VO estimate and reformulating the map landmark uncertainties in the MEL algorithm to account for dependence between map landmarks. To validate these uncertainties, a field test will need to be conducted to prove that uncertain experiences with poor localization no longer cause lock in. A detailed discussion on this topic can be found in Section 5.6.

Apart from preventing locking in to bad experiences, an estimator with realistic uncertainties can be used to autonomously stop the robot if it is too uncertain of its position. The robot can then use its current experience to repeat back to a previous position in the network of paths and either return to the starting point or replan an alternative route. The current system stops the robot if the uncertainties grow beyond a fixed threshold throughout the entire network. Recent work from Dequaire et al. (2016) present a method to predict the lateral bounds around the taught path where the robot is able to likely localize by examining visual data from the traverse. This information can be used in the VT&R 2.0 system to provide place-unique uncertainty thresholds to stop the robot faster in areas where localization is difficult.

As discussed in Section 5.6, it is unclear to what degree of viewpoint change the VT&R 2.0 system can tolerate during an autonomous traverse. To this end, further analysis on the performance of the system with respect to viewpoint change is warranted. This can be achieved through performance analyses on public datasets that contain viewpoint and appearance change such as the Oxford 1000 km dataset (Maddern et al., 2017) and field tests where the rover is occasionally driven offset from the taught path.

The MEL algorithm relies on the ability to bridge the appearance gap between the live and privileged experience through autonomous experiences in the map. The results of the field tests, presented in Section 5.5 demonstrated the system's ability to reliably localize the live experience to the privileged experience across extreme appearance change when there are a sufficient amount of bridging experiences in the map. This requires a system whose autonomy schedule is dependent on the rate of appearance change. While this aspect is the inherent nature of the MEL localization scheme, the number of experiences needed to capture appearance change can be reduced. One simple enhancement to mitigate this issue is to incorporate the multi-stereo localization system presented in Chapter 3 to expand the field of view of the localizer and increase the overall number of feature matches observed. Another strategy is through the use of visual features that are less susceptible to appearance change. Potential candidates for use in the MEL framework are the SVM-based scene signatures, (McManus et al., 2014b; Linegar et al., 2016), and visual features trained to be less variant to appearance change (Krajník et al., 2016). These additional feature types can be used in conjunction with traditional point-based visual features in the VT&R 2.0 system's multi-channel localization framework. Besides point-based visual feature enhancements, performance across appearance change in the MEL framework could be improved through the adoption of dense localization techniques (Wolcott and Eustice, 2015; Pascoe et al., 2017) which will require significantly more research to incorporate as it is not readily apparent how to integrate these methods into the MEL algorithm.

The primary motivation behind developing a vision-only system is the commercial ubiquity and low price point of high-quality passive camera sensors. However, if price of the sensor is not a concern, then autonomous path following with active sensors is a promising avenue of research. Preliminary results in Cornick et al. (2016) have shown success in localization with a ground-penetrating RADAR. This could be used in conjunction with a stereo VO pipeline in the VT&R 2.0 framework to achieve true appearance-invariant navigation.

These possibilities for future work will further improve the robustness of long-term autonomous path-following methods as well as metric localization across extreme appearance change.

Bibliography

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W. (2013). Robust map optimization using dynamic covariance scaling. In *2013 IEEE International Conference on Robotics and Automation*, pages 62–69.

Agrawal, M., Konolige, K., and Blas, M. R. (2008). *CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching*, pages 102–115. Springer Berlin Heidelberg, Berlin, Heidelberg.

Anderson, S. and Barfoot, T. (2015). Full steam ahead: Exactly sparse gaussian process regression for batch continuous-time trajectory estimation on se(3). In *Intelligent Robots and Systems (IROS)*.

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Barfoot, T. D. and Furgale, P. T. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359.

Berczi, L.-P. and Barfoot, T. D. (2016). It's like déjà vu all over again: Learning place-dependent terrain assessment for visual teach and repeat. In *IROS*.

Biber, P. and Duckett, T. (2005). Dynamic maps for long-term operation of mobile service robots. In *In Proc. of Robotics: Science and Systems (RSS)*.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). *BRIEF: Binary Robust Independent Elementary Features*, pages 778–792. Springer Berlin Heidelberg, Berlin, Heidelberg.

Chen, Z. and Birchfield, S. T. (2009). Qualitative vision-based path following. *IEEE Trans. on Robotics*, 25(3):749–754.

Chum, O., Matas, J., and Kittler, J. (2003). Locally optimized ransac. In *Pattern recognition*, pages 236–243. Springer.

Churchill, W. and Newman, P. (2013). Experience-based navigation for long-term localisation. *The Int. Journal of Robotics Research*, 32(14):1645–1661.

Clipp, B., Kim, J.-H., Frahm, J.-M., Pollefeys, M., and Hartley, R. (2008). Robust 6DOF Motion Estimation for Non-Overlapping, Multi-Camera Systems. In *Proceedings of the 2008 IEEE Workshop on Applications of Computer Vision*, Washington, DC, USA.

Corke, P., Paul, R., Churchill, W., and Newman, P. (2013). Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*.

Cornick, M., Koechling, J., Stanley, B., and Zhang, B. (2016). Localizing ground penetrating radar: A step toward robust autonomous ground vehicle localization. *Journal of Field Robotics*, 33(1):82–102.

Cummins, M. and Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665.

Dequaire, J., Tong, C. H., Churchill, W., and Posner, I. (2016). Off the beaten track: Predicting localisation performance in visual teach and repeat. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden.

Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*.

Finlayson, G., Hordley, S., Cheng, L., and Drew, M. (2006). On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395.

Furgale, P. and Barfoot, T. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560.

Furgale, P. and Tong, C. (2010). Speeded up speeded up robust features [online]. Avaliable: http://asrl.utias.utoronto.ca/code/gpusurf/. [Accessed: 3- March- 2016].

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151.

Heng, L., Lee, G. H., and Pollefeys, M. (2014). Self-calibration and visual slam with a multi-camera system on a micro aerial vehicle. In *Proceedings of Robotics: Science and Systems (RSS)*.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

Johnson, A. E., Goldberg, S. B., Cheng, Y., and Matthies, L. H. (2008). Robust and efficient stereo feature tracking for visual odometry. In *2008 IEEE International Conference on Robotics and Automation*, pages 39–46.

Kazik, T., Kneip, L., Nikolic, J., Pollefeys, M., and Siegwart, R. (2012). Real-time 6D stereo visual odometry with non-overlapping fields of view. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Klein, G. and Murray, D. (2007). Parallel Tracking and Mapping for Small AR Workspaces. In *Int. Symposium on Mixed and Augmented Reality*, pages 1–10.

Kneip, L., Furgale, P., and Siegwart, R. (2013). Using multi-camera systems in robotics: Efficient solutions to the npnp problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Konolige, K. and Bowman, J. (2009). Towards lifelong visual maps. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1156–1163.

Krajník, T., Cristóforis, P., Kusumam, K., Neubert, P., and Duckett, T. (2016). Image features for visual teach-and-repeat navigation in changing environments. *Robotics and Autonomous Systems*.

Krajnk, T., Faigl, J., Vonsek, V., Konar, K., Kulich, M., and Peuil, L. (2010). Simple yet stable bearing-only navigation. *JFR*, 27(5):511–533.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Krüsi, P., Bücheler, B., Pomerleau, F., Schwesinger, U., Siegwart, R., and Furgale, P. (2014). Lighting-Invariant Adaptive Route Following Using ICP. *Journal of Field Robotics*, 32(4):534–564.

Lee, G. H., Faundorfer, F., and Pollefeys, M. (2013). Motion estimation for self-driving cars with a generalized camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, Oregon, USA.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Linegar, C., Churchill, W., and Newman, P. (2015). Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation. In *ICRA*.

Linegar, C., Churchill, W., and Newman, P. (2016). Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden.

Liu, N. N., Zhao, M., Xiang, E., and Yang, Q. (2010). Online evolutionary collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 95–102, New York, NY, USA. ACM.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Lowry, S. and Milford, M. J. (2016). Supervised and unsupervised linear learning techniques for visual place recognition in changing environments. *IEEE Transactions on Robotics*, 32(3):600–613.

Lowry, S., Wyeth, G., and Milford, M. (2012). Cat-graph+ : towards odometry-driven place consolidation in changing environments. In Carnegie, D., editor, *2012 Australasian Conference on Robotics and Automation*, Wellington, New Zealand. Australian Robotics & Automation Association.

MacTavish, K. and Barfoot, T. (2014). Towards hierarchical place recognition for long-term autonomy. In *ICRA Workshop*.

MacTavish, K. and Barfoot, T. D. (2015). At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Computer and Robot Vision (CRV), 2015 12th Conference on*, pages 62–69.

MacTavish, K. and Barfoot, T. D. (2017). Night rider: Visual odometry using headlights. In *Computer and Robot Vision (CRV), 2017 14th Conference on*.

Mactavish, K., Michael, P., and Barfoot, T. D. (2018). Selective Memory: Recalling Relevent Experience for Long-Term Visual Localization. *The International Journal of Robotics Research, in preparation*.

MacTavish, K., Paton, M., and Barfoot, T. (2015). Beyond a shadow of a doubt: Place recognition with colour-constant images. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, Toronto, Ontario, Canada.

MacTavish, K., Paton, M., and Barfoot, T. (2017). Visual triage: A bag-of-words experience selector for long-term visual route following. In *ICRA*.

Maddern, W., Milford, M., and Wyeth, G. (2012a). Cat-slam: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The Int. Journal of Robotics Research*, 31(4):429–451.

Maddern, W., Milford, M., and Wyeth, G. (2012b). Towards persistent indoor appearance-based localization, mapping and navigation using cat-graph. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 4224–4230.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15.

Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging Experience for Large-Scale LIDAR Localisation in Changing Cities. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, Seattle, WA, USA.

Maddern, W., Stewart, A., and Newman, P. (2014). LAPS-II: 6-DoF day and night visual localisation with prior 3D structure for autonomous road vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium*.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics*, 24(3):169–186.

Matthies, L. H. (1989). *Dynamic Stereo Vision*. PhD thesis, Pittsburgh, PA, USA. AAI9023429.

McManus, C., Churchill, W., Maddern, W., Stewart, A., and Newman, P. (2014a). Shady dealings: Robust, long- term visual localisation using illumination invariance. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China.

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. (2012). Visual teach and repeat using appearance-based lidar. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, St. Paul, Minnesota, USA.

McManus, C., Upcroft, B., and Newman, P. (2014b). Scene signatures: Localised and point-less features for localisation. In *Robotics Science and Systems (RSS)*.

McManus, C., Upcroft, B., and Newman, P. (2015). Learning place-dependant features for long-term vision-based localisation. *Autonomous Robots*, 39(3):363–387.

Milford, M. and Wyeth, G. (2012). Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*.

Moravec, H. P. (1980). *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, Stanford, CA, USA. AAI8024717.

Muhlfellner, P., Brki, M., Bosse, M., Derendarz, W., Philippsen, R., and Furgale, P. (2015). Summary maps for lifelong visual localization. *Journal of Field Robotics*, pages n/a–n/a.

Naseer, T., Ruhnke, M., Spinello, L., Stachniss, C., and Burgard, W. (2015). Robust visual slam across seasons. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*.

Naseer, T., Spinello, L., Burgard, W., and Stachniss, C. (2014). Robust visual robot localization across seasons using network flows. In *Proc. of the Conf. on Artificial Intelligence*.

Nelson, P., Churchill, W., Posner, I., and Newman, P. (2015). From Dusk till Dawn: Localisation at Night using Artificial Light Sources. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA.

Neubert, P., , N., and Protzel, P. (2013). Appearance change prediction for long-term navigation across seasons. In *Proceedings of the European Conference on Mobile Robots (ECMR)*.

Oskiper, T., Zhu, Z., Samarasekera, S., and Kumar, R. (2007). Visual odometry system using multiple stereo cameras and inertial measurement unit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ostafew, C. J., Schoellig, A. P., Barfoot, T. D., and Collier, J. (2016). Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking. *Journal of Field Robotics*, 33(1):133–152.

Otsu, K., Otsuki, M., and Kubota, T. (2015). Experiments on stereo visual odometry in feature-less volcanic fields. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*.

Pascoe, G., Maddern, W., and Newman, P. (2015a). Robust Direct Visual Localisation using Normalised Information Distance. In *British Machine Vision Conference (BMVC)*, Swansea, Wales.

Pascoe, G., Maddern, W., Stewart, A. D., and Newman, P. (2015b). FARLAP: Fast Robust Localisation using Appearance Priors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA.

Pascoe, G., Maddern, W., Tanner, M., Pinies, P., and Newman, P. (2017). NID-SLAM: Robust monocular SLAM using normalised information distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI.

Paton, M., MacTavish, K., Berczi, L., van Es, K., and Barfoot, T. (2017a). I can see for miles and miles: An extended field test of visual teach & repeat 2.0. In *Proc. of Field and Service Robotics (FSR), to appear.*

Paton, M., MacTavish, K., Warren, M., and Barfoot, T. (2016). Bridging the appearance gap: Multi-experience localization for long-term visual teach & repeat. In *IROS.*

Paton, M., McTavish, K., Ostafew, C., and Barfoot, T. (2015a). It's not easy seeing green: Lighting-resistant visual teach & repeat using color-constant images. In *Proc of the Int. Conf. Robotics and Automation (ICRA).*

Paton, M., Pomerleau, F., and Barfoot, T. (2015b). Eyes in the back of your head: Robust visual teach & repeat using multiple stereo cameras. In *Proc. of the Conf. on Computer and Robot Vision (CRV).*

Paton, M., Pomerleau, F., and Barfoot, T. (2015c). In the dead of winter: Challenging vision-based path following in extreme conditions. In *Proc. of Field and Service Robotics (FSR).*

Paton, M., Pomerleau, F., MacTavish, K., Ostafew, C. J., and Barfoot, T. D. (2017b). Expanding the limits of vision-based localization for long-term route-following autonomy. *Journal of Field Robotics*, 34(1):98–122.

Pepperell, E., Corke, P., and Milford, M. (2015). Automatic image scaling for place recognition in changing environments. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA).*

Pless, R. (2003). Using many cameras as one. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Powell, M. J. D. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155.

Raguram, R., Chum, O., Pollefeys, M., Matas, J., and Frahm, J. M. (2013). Usac: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038.

Ratnasingam, S. and Collins, S. (2010). Study of the photodetector characteristics of a camera for color constancy in natural scenes. *J. Opt. Soc. Am. A*, 27(2):286–294.

Rawlings, J. and Mayne, D. (2009). *Model Predictive Control: Theory and Design.* Nob Hill Pub.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571.

Stenning, B. E., McManus, C., and Barfoot, T. D. (2013). Planning using a network of reusable paths: A physical embodiment of a rapidly exploring random tree. *Journal of Field Robotics*, 30(6):916–950.

Sunderhauf, N., Neubert, P., and Protzel, P. (2013). Are we there yet? challenging seqslam on a 3000 km journey across all four seasons.

Tribou, M. J., Harmat, A., Wang, D. W., Sharf, I., and Waslander, S. L. (2015). Multi-camera parallel tracking and mapping with non-overlapping fields of view. *The International Journal of Robotics Research*, 34(12):1480–1500.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380.

Van Es, K. and Barfoot, T. (2015). Being in two places at once: Smooth visual path following on globally inconsistent pose graphs. In *Proceedings of the 12th Conference on Computer and Robot Vision (CRV)*, Halifax, Nova Scotia, Canada.

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

Wolcott, R. W. and Eustice, R. M. (2014). Visual localization within LIDAR maps for automated urban driving. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, Chicago, IL, USA.

Wolcott, R. W. and Eustice, R. M. (2015). Fast LIDAR localization using multiresolution Gaussian mixture maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seattle, WA, USA. Accepted, To Appear.