Visual Teach and Repeat Using Appearance-Based Lidar
— A Method For Planetary Exploration —

by

Colin McManus

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

Abstract

Visual Teach and Repeat Using Appearance-Based Lidar

— A Method For Planetary Exploration —

Colin McManus

Master of Applied Science

Faculty of Engineering, Institute for Aerospace Studies

University of Toronto

2011

Future missions to Mars will place heavy emphasis on scientific sample and return operations, which will require a rover to revisit sites of interest. Visual Teach and Repeat (VT&R) has proven to be an effective method to enable autonomous repeating of any previously driven route without a global positioning system. However, one of the major challenges in recognizing previously visited locations is lighting change, as this can drastically change the appearance of the scene. In an effort to achieve lighting invariance, this thesis details the design of a VT&R system that uses a laser scanner as the primary sensor. The key novelty is to apply appearance-based vision techniques traditionally used with camera systems to laser intensity images for motion estimation. Field tests were conducted in an outdoor environment over an entire diurnal cycle, covering more than 11km with an autonomy rate of 99.7% by distance.

Acknowledgements

I owe my entire academic career as a researcher to Professor Timothy Barfoot — my supervisor and mentor. Without his guidance and wisdom, as well as his infectious, relentless drive to be the very best in our field, I would not have had the privilege to conduct such world-class, cutting-edge research. I am forever indebted and honoured to have been one of his students.

I also owe a great deal of thanks to my colleague Paul Furgale, who has taught me so much over these past two years and played a large role in all of the work I have done; whether it was programming assistance, paper critiques, or lessons on state-of-the-art estimation techniques. I was fortunate to have the benefit to work with such an experienced and incredible computer scientist and a wonderful person.

Lastly, I wish to acknowledge my loving friends and family for their continued support throughout my academic studies. Without them I would not be where I am today.

Thank you all so much.

Colin McManus

September 14, 2011

Contents

List of Tables

List of Figures

Notation

Symbol	Description
x	A real-valued scalar
$\mathbf{x}$	A real-valued $N \times 1$ column vector
$\underrightarrow{\mathcal{F}}$	A reference frame *vectrix* defined by three unit vectors
$\mathbf{C}_{a,b}$	A 3×3 rotation matrix from frame 'b' to frame 'a'
$\boldsymbol{\rho}_a^{b,a}$	A vector pointing from frame 'a' to frame 'b' and expressed in frame 'a'
$\mathbf{T}_{a,b}$	A 4×4 transformation matrix from frame 'b' to frame 'a'
$\mathbf{1}$	The identity matrix
$\mathbf{0}$	The zero matrix
$\sim \mathcal{N}(\mathbf{x}, \mathbf{P})$	Normally distributed with mean $\mathbf{x}$ and covariance $\mathbf{P}$
$\mathcal{I}$	A stack of $N \times N$ intensity, azimuth, elevation, and range images

Chapter 1

Introduction

1.1 The Quest for Knowledge

Increasing autonomy for planetary rovers has been an ongoing effort for decades. However, current rover technologies for planetary exploration still lack the sufficient level of autonomy required for many navigation tasks and, in most cases, require human-in-the-loop interaction. This lack of autonomy is the main reason the Mars Exploration Rovers were limited to traverses of less than 40m per sol (Biesiadecki et al., 2006). This research is primarily focused on developing autonomous rover technologies in order to maximize the scientific return and ultimately help further our understanding of the nature and history of the universe.

Future mission concepts to Mars will place heavy emphasis on scientific sample and return missions, which have been in development for many years (Bajracharya et al., 2008; Schenker et al., 2003). Mars sample-and-return missions are based on the concept of repeated in-field retrieval of samples by a rover and subsequent rendezvous with an

Figure 1.1: Mars sample and return concept.

Earth return vehicle (Bajracharya et al., 2008). Visual Teach and Repeat (VT&R) has proven to be an effective technology for enabling a vehicle to autonomously repeat a previously driven route and thus lends itself very well to sample and return missions. The basic strategy behind VT&R is the following. During a *teach pass*, the vehicle uses a visual sensor to construct a series of maps that are stored in memory. The *repeat pass* is accomplished by referencing these archived maps and comparing current views with the previously seen views in order to autonomously retrace the taught route. Furgale and Barfoot (2010) demonstrated the success of this VT&R approach through long-range field trials in a planetary analogue environment. In total, over 32km were traversed and the robot was able to repeat the previous routes with a 99.6% autonomy rate by distance. Although very effective, the use of a stereo camera as the primary sensor proved to be one of the fundamental limitations of the system, as changes in ambient lighting would sometimes result in a failure to recognize a pre-visited location.

Stereo cameras have emerged as the dominant sensor modality in outdoor, unstructured terrain, largely due to the success of sparse, appearance-based computer vision techniques. However, the 'Achilles heel' for all camera-based systems is their dependence on ambient lighting. This dependence poses a serious problem in outdoor environments that lack adequate or consistent light, such as the Moon. Even on Earth, lighting changes over the course of a day can result in failures to recognize pre-visited areas (Furgale and Barfoot, 2010). Some researchers have examined using external light sources to illuminate dark environments. For example, Husmann and Pedersen (2008) considered dark stereo navigation aided by LED spotlights for illumination. Their method combined low-dynamic range (LDR) images at various exposures to create high-dynamic range (HDR) images that are used for stereo. Although they demonstrated that ranging accuracy between sunlit and LED-lit images could be similar depending on the amount of texture in the scene, they concluded that for continuous motion, specialized camera hardware and high-power lighting (1 kW) would be required just to achieve a maximum lookahead range of 10m. Even if such a power budget were available, scene appearance from night to day could be drastically different, which would not work for VT&R.

(a) Camera intensity image.

(b) Processed lidar intensity image.

Figure 1.2: Raw camera image and a processed lidar intensity image. Image credit: McManus et al. (2011).

Unlike cameras, active sensors such as light detection and ranging (lidar) sensors, use their own light source to illuminate the scene, making them a favourable alternative in light-denied environments such as the Moon. Wettergreen et al. (2009) have developed a lunar rover prototype called Scarab, which uses an onboard lidar to navigate in the dark. Their approach estimates motion based on the classic Iterative Closest Point (ICP) scan matching algorithm (Besl and McKay, 1992), which determines the transformation that best aligns two point clouds, in the least-squares sense. ICP-based approaches are ubiquitous for systems that use laser scanners and have been successfully used for localization and mapping in planar 3D environments (Wulf et al., 2008; Surmann et al., 2003; Thrun et al., 2000) and outdoor environments (Wettergreen et al., 2009; Rekleitis et al., 2007; Se et al., 2004). However, for dense 3D data, scan matching techniques are too computationally intensive to be run online, require a good initial guess to ensure convergence, and are heavily dependent on distinctive topography.

This thesis presents an appearance-based VT&R system that uses lidar as the primary sensor and bridges the gap between the efficiency of appearance-based techniques and the lighting invariance of 3D laser scanners by applying appearance-based methods to 2D lidar intensity images[1]. These lidar intensity images look nearly identical to a standard grayscale camera image, but with the added benefit of looking the same both in the light and in the dark (see Figure 1.2 for an example of a camera/lidar intensity image). Combined with the azimuth,

[1] 3D laser scanners use rotating mirrors to steer a laser beam across a raster pattern in azimuth and elevation. In contrast, 2D laser scanners only scan in one plane.

elevation, and range data, a lidar provides all the necessary appearance and metric information required for motion estimation. This lidar-based VT&R system is a first of its kind and has been validated and tested in a planetary analogue environment, autonomously repeating over 11km in a variety of lighting conditions. The system only relies on frame-to-frame Visual Odometry (VO) using sparse bundle adjustment, but is able to repeat routes accurately and consistently with root-mean-squared path errors on the order of centimeters. The key to the technique's accuracy is its use of a relative map representation combined with a novel and efficient *sliding local map* approach for localizing against the map. This approach obviates the need for a complex Simultaneous Localization and Mapping (SLAM) system (Durrant-Whyte and Bailey, 2006), as it is both efficient and effective for navigation tasks that require revisiting previous locations.

1.2 Contributions

This thesis has resulted in a number of novel contributions to the field of mobile robotics. In particular, work leading up to this thesis led to a publication that demonstrated how appearance-based methods can be successfully applied to scanning laser-rangefinders for visual odometry (McManus et al., 2011). This was an important stepping stone for the work presented in this thesis, as it validated the motivation for using a 3D lidar sensor in place of a camera.

The following is a list of novel contributions that have resulted from this thesis work.

1. The first to use a laser scanner for appearance-based VO (McManus et al., 2011),
2. The first VT&R system that uses appearance-based lidar to achieve lighting invariance,
3. The development of a novel *sliding local map* approach for matching against the map during the repeat pass (this approach embeds information from neighbouring keyframes into an augmented keyframe for improved accuracy),
4. Experimental validation of the algorithm in a planetary analogue environment, traversing over 11km almost fully autonomously.

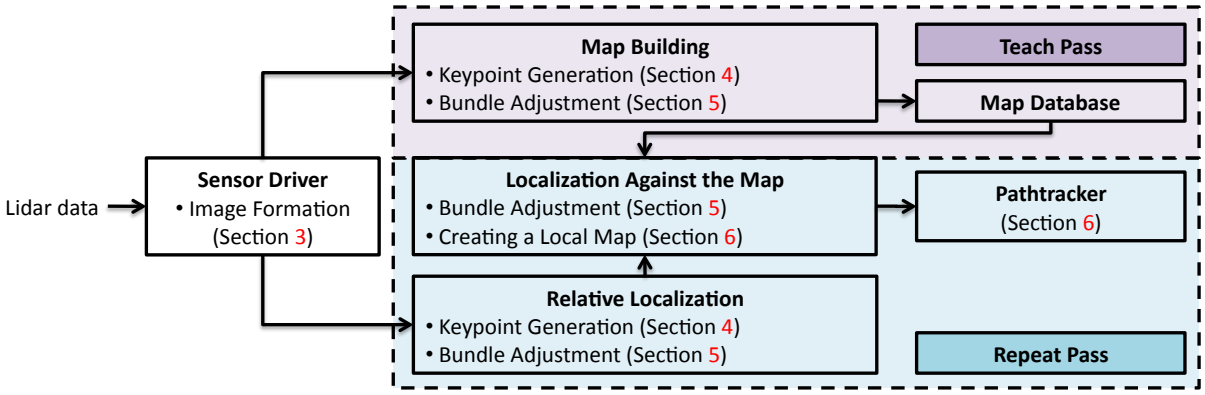


Figure 1.3: High-level block diagram of the major code blocks in the VT&R system.

1.3 High-level Overview

This thesis is structured according to the high-level system diagram illustrated in Figure 1.3. This figure shows the general layout of the VT&R architecture, which is composed of two phases: a *teach pass* and a *repeat pass*. Critical elements of the system are shown in bullets under some of the major code blocks. These critical elements are: (i) image formation (Section 3), (ii) keypoint generation (Section 4), (iii) bundle adjustment (Section 5), and (iv) creating the local map and path tracking (Section 6). Following this as an outline, Section 2 will begin with a review of previous VT&R systems discussed in the literature as well as related work using lidar intensity images for motion estimation. Section 3 provides a description of the image formation process, which involves rendering 2D laser intensity images from raw lidar data and forming an *image stack* of intensity, azimuth, elevation, and range data. Section 4 describes how keypoints are formed using the image stack as well the keypoint matching and outlier rejection process used for motion estimation. Section 5 provides a mathematical treatment of the estimation theory used for this VT&R system, which is based on batch bundle adjustment. The local map approach is discussed in section 6 along with a more detailed system architecture. Lastly, section 7 presents results for a series of long-range VT&R experiments, which involved teaching a route during daylight outdoors and autonomously repeating the same route every 2-3 hours for over 25 hours.

Chapter 2

Related Work

This chapter provides a review of related work on VT&R systems, categorizing them according to their map representation, which is a critical design decision as it affects both the system's efficiency and performance. The current trend in robotics appears to be moving away from the long-held belief that everything must be estimated in a single privileged coordinate frame, as relative map representations can be just as effective and are less computationally expensive (e.g., Sibley et al. (2010)).

Lidar-based localization methods are also discussed, with particular emphasis on techniques that utilize lidar intensity images for motion estimation. At present, only May et al. (2009), Ye and Bruch (2010), and McManus et al. (2011) have used lidar intensity images for motion estimation in the mobile robotics literature. However, it should be noted that both May et al. (2009) and Ye and Bruch (2010) used a Swiss Ranger time-of-flight (TOF) camera, which is equipped with an array of LEDs to simultaneously illuminate the scene. This is in contrast to laser scanners, which illuminate the environment with a single light source, introducing new problems such as image formation and image distortion caused by moving and scanning at the same time. To date, McManus et al. (2011) are the only ones who have used laser scanners for appearance-based motion estimation.

2.1 Visual Teach and Repeat

One of the main distinguishing features between various VT&R methods is their map representation, which can be categorized as either metric, topological, or a hybrid of the two. Metric maps are fine-grained representations of the environment (e.g., monolithic, globally consistent map), whereas topological maps are graph-like with nodes representing significant regions or landmarks and edges representing interconnections between places (Thrun et al., 2005). Topological/metric maps are hybrid representations where the nodes can be locally consistent metric maps (Simhon and Dudek, 1998). The following is a review of VT&R systems according to their map representation.

Metric Map Representations

Baumgartner and Skaar (1994) developed a VT&R system for structured environments that used a monocular camera as the primary sensor. They placed ring-shaped visual cues throughout the environment and fused wheel odometry with monocular observations of these visual cues to generate a map during the route teaching (the Extended Kalman Filter was used for sensor fusion). Route following was accomplished in the same manner, by localizing against the archived map using both wheel odometry and the monocular observations of the visual landmarks. Richardson and Rodgers (2001) developed a VT&R concept for the military, in which a robot would track and follow a soldier and autonomously repeat the taught route, carrying supplies or munitions back and forth. During the teaching phase, a voxelized[1] route map would be generated and stored in a modified octree data structure[2]. This route map was made to be globally consistent; however, the authors recognized that locally consistent maps would have been sufficient. Although the system was never implemented, simulations demonstrated the possibility of a robot moving at walking speeds during the repeat pass.

[1] A voxel, or *volumetric pixel*, is a volume element in a three dimensional grid.
[2] A tree data structure where each node has exactly eight children.

Kidono et al. (2002) developed a stereo-based VT&R system for indoor environments that used an active localization strategy to point their sensor head during the repeat pass. Their system searched for features at structure boundaries, called *object points*, in order to create a 2D range profile of the environment and construct a 2D grid-based map. After the teaching phase, a minimum-distance path to the starting position would be planned, making their system more flexible than most other VT&R systems. However, in order to ensure that they could still successfully localize against the archived map, the stereo camera was actively directed towards the closest observable landmark in the map. If the closest landmark was not visible (because of an obstruction), then they pointed their sensor towards the landmark that minimized the area of the robot's uncertainty ellipse.

Royer et al. (2007) presented an outdoor VT&R system that used a monocular camera for visual input. They used the Harris corner detector (Harris and Stephens, 1988) to detect keypoints in image space and triangulated these keypoins to 3D landmarks, which were embedded in a global map (this map building was a batch process that was done offline). To resolve the scale ambiguity inherent to monocular navigation, they manually entered the length of the path traveled. During the autonomous repeating, the system would search for the closest keyframe in the map and transform the landmarks observed in this closest keyframe to keypoints, which would then be matched against the current image. Like this system, the approach presented in this thesis matches against the nearest keyframe during the repeat pass; however, there are a number of significant differences that warrant discussion. First, Royer et al. (2007) embed all of the landmarks in a single coordinate frame, whereas the system in this thesis preserves the keypoints in each local keyframe's reference frame. Second, this system only matches against a single keyframe, whereas the system described in this thesis creates an *augmented keyframe* that contains keypoints over a window of keyframes for improved accuracy and robustness. Third, the system described in this thesis performs the map building online, as there is no requirement to construct a globally consistent map. Fourth, although their system demonstrated centimeter-level localization accuracy, their outdoor experiments were restricted to small tra-

verses on the order of 130m and on paved roads. In contrast, the experiments presented in this thesis were conducted over kilometers of challenging 3D, unstructured terrain.

Topological Map Representations

It is worthwhile to preface this section with a brief discussion on the difference between *visual homing* and topological-based VT&R systems. In visual homing (VH) the goal is to return to a home position by only matching the current view with a single snapshot of the goal location, meaning that one must start within close proximity to the goal (e.g., see Vardy and Oppacher (2003); Franz et al. (1998) for representative techniques). In VT&R, the goal is to repeat a previously driven route by localizing against a series of archived maps (e.g., images). VH is based on the snapshot model proposed by Cartwright and Collett (1987, 1983), which was inspired by honeybee search patterns. Clearly, for longer distances, this simplified VH method will simply fail due to significant view-point changes. As a result of this shortcoming, a number of variants of VH emerged, which appended the path with multiple *home* locations (e.g., Bekris et al. (2006); Argyros et al. (2005)). Thus, one could argue that as the number of target locations increases, these methods will essentially converge to topological-based VT&R systems.

Matsumoto et al. (2000, 1996) developed what they called a *view-sequenced route representation* (VSRR), which are low resolution images captured by a camera during a teach pass and stored in memory. In this VSRR, the interval between successive images was variable and depended on scene complexity and memorization thresholds. For the repeat pass, a cross-correlation procedure was performed on the current image and the next candidate image in the VSRR to determine if the vehicle moved into the next local map. Arguing that correlation correspondence over the entire image was not efficient, Ohno et al. (1996) presented a different approach for matching against the map when using the VSRR. Instead of using the entire image, they extracted vertical lines from pre-recorded and current camera images in order to determine correspondences (i.e., they used vertical lines as features). Jones et al. (1997) presented work on a fully integrated navigation system that used the VSRR for appearance-

based navigation, but instead of template matching over the entire image, they also opted for a different matching approach. More specifically, they used zero-mean energy-normalized cross-correlation for matching current views to archived views. Matsumoto et al. (1999) and Tang and Yuta (2001) extended the VSRR approach to omnidirectional cameras, which reportedly improved the system's localization performance.

Blanc et al. (2005) presented an indoor VT&R system that stored key images during the teach pass according to matching thresholds, making this very similar to the VSSR by Matsumoto et al. (1996). They pointed a conventional camera towards the ceiling to extract features during the teaching phase, which also provided a good normal approximation for their homography matrix. For the repeat pass, they compared current images with key images and used a 2.5D visual servoing control law (Mails et al., 1999) to guide the robot along the route. This approach was adapted to perspective and catadioptric cameras by Courbon et al. (2007)[3].

Goedeme et al. (2005) described a VT&R technique that used omnidirectional images taken at sparse distance intervals of 2-4m. Their visual homing operations consisted of two phases: an initialization phase (wide baseline feature correspondences, generation of a homing vector, local map estimation) and an update phase (feature tracking, local map updating, homing vector updating). An extension of this work was presented by Goedeme et al. (2007), where a visual servoing control scheme was used to steer the vehicle during the repeat pass and Dempster-Shafer (Dempster, 1967; Shafer, 1976) theory of evidence was used to aid in detecting loop closures in the topological map. The only sensor used was an omnidirectional colour camera. Based on this work, Fraundorfer et al. (2007) presented a similar approach that performed map building online, used shorter distances between images, and also considered the problem of path planning after the teach pass. They used an efficient compact image representation of *visual words*, adopted from Nistér and Stewénius (2006). Using the 5-point algorithm (Nistér, 2003), relative orientations were computed from the feature correspondences between images,

[3]A catadioptic camera system uses an upward looking camera with a hyperbolodial mirror mounted above it (Goedeme et al., 2007).

which were then used for navigation. Path planning was accomplished with a graph-based planner, the details of which were not discussed in their paper.

Bekris et al. (2006); Argyros et al. (2005, 2001) introduced a VT&R method that used a bearing-only control law for autonomous route repeating and a panoramic camera as the primary sensor. Their controller requires at least three matched features between panoramas, but makes no assumption regarding the robot's orientation. In their experiments, the image separation was very large and lead to non-identical teach-and-repeat paths; however, they were still able to home the vehicle with reasonable accuracy. A similar bearing-only control strategy was used by Booij et al. (2007). However, instead of maintaining a directional graph structure between nodes, they exhaustively matched all archived images to create a graph with links connecting all matching images. Each link is assigned a weighting that is based on the number of matches between the new images (called a *similarity score*). Given a goal location, their system would plan a path to the goal according to the links that have the highest similarity score. They used epipolar geometry with a planar floor constraint to calculate the heading between two images, and used a control strategy that simply directed the robot from one node to the next. Chen and Birchfield (2006) developed a VT&R system that used a Kanade-Lucas-Tomasi (KLT) feature tracker (Tomasi and Kanade, 1991; Lucas and Kanade, 1981) and a qualitative control scheme that acted as a bang-bang controller for robot motions. In other words, the controller used feature positions from the current/destination images as input and the output consisted of 1 out of 3 possible commands: continue straight, turn left, or turn right.

Diosi et al. (2007) described a VT&R approach in outdoor environments that used only a monocular camera. They considered overlapping local paths, and used local 3D information, contrast compensation, and visual servoing for autonomous playback. Since visual servoing does not depend on an accurate estimate of the robot's pose, this technique allowed for larger 3D reconstruction error, which in turn, allowed for greater distances between image pairs and less memory usage. This work was extended by Šegvić et al. (2009), with the inclusion of a feature prediction step to account for lost features due to occlusions.

Koch et al. (2010); Koch and Teller (2009) developed a topological VT&R system that works in locally planar environments and does not require any knowledge of the extrinsic/intrinsic camera parameters. Instead, they use an automated training system to learn the geometric relationships needed to compute the heading between subsequent images, which only needs to be done once for a given camera configuration (these geometric relationships are encoded in what they call a *match matrix*). During the teach pass, they build a topological network of nodes that contain Scale Invariant Feature Transform (SIFT) keypoints. A *bag of words* place recognition system similar to Cummins and Newman (2008) is used to detect loop closures. During the repeat pass, they maintain a probability density of the topological position of the vehicle over a window of nodes, modelling the system as a first-order Markov process. The state transitions are modelled as a Gaussian over a window of nodes and the observation model is related to a visual similarity score between images. Heading-angle control commands are based purely on the orientation error between adjacent images, which is computed as follows. Feature matches between subsequent images are determined using the nearest neighbour method in descriptor space and for each match, their distance in image space is translated into rotational displacement based on the match matrix. They demonstrated this system in indoor office environments covering more than 1.2km. However, this system only provides rough bounds and estimates on the orientation and as a result, the repeated path often deviated from the taught path. Thus, this system would not be suitable in outdoor, unstructured terrain, nor would it be able to cope with severe lighting changes since it uses a camera for visual input.

Topological/Metric Map Representations

Marshall et al. (2008) developed a lidar-based VT&R method for autonomous underground tramming in planar environments using a SICK laser. For mapping during the teaching phase, they used a variant of the Atlas framework (Bosse et al., 2004), which is a sequence of overlapping metric maps attached to the path. The maps overlap in order to ensure a smooth transition from one map to another. After each teach pass, offline route profiling was performed, which

generated a series of overlapping local maps, speed profiles, and a record of any pause points. During the playback phase, the vehicle would repeat the path as dictated in the route profile.

Zhang and Kleeman (2009) developed a VT&R system for planar environments that used an omnidirectional camera as the primary sensor. During the teaching phase the robot was manually piloted along a route, recording images spaced out every 35cm and/or 5° according to odometry. During the repeat pass, a window of archived images were matched against the most recent image using image cross-correlation (ICC), which was done in the Fourier domain for efficiency. Over 18km of experiments were presented in both indoor/outdoor planar environments, demonstrating good performance overall, with errors on the order of 10cm and with only a few failures due to orientation tracking errors and map localization errors. It should be noted that for groundtruth, they would simply stop the robot at waypoints and measure the translational offsets (i.e., GPS was not used and this error measure only took place at discrete sections along the path).

Furgale and Barfoot (2010) were the first to develop a fully 3D VT&R system for outdoor, unstructured environments and validated the system in the Canadian High Arctic. The route teaching involved capturing and logging stereo images for post-processing into a series of locally consistent overlapping maps, where 3D landmarks were embedded within each local map. Route repeating used a route manager to handle map switching, vehicle speed control, and error monitoring. A failure handling module was used, which stopped the robot once it passed a certain distance without making a successful global localization. Recovery was achieved by searching the image database for a map that matched the current location, after which, the route repeating phase would be re-initialized. For localization, their system would interleave visual odometry (VO) with localizing against the map, which was done for computational reasons as they were unable to perform both within the same control cycle. The system was field tested in Devon Island, and of the 32.919km traveled, 99.6% was traversed autonomously.

Like Furgale and Barfoot (2010), the VT&R system presented in this thesis falls under the topological/metric map category, since the map is a topological network of keyframes that

contain metric information. What differentiates the system described in this thesis with the one from Furgale and Barfoot (2010) is the following: (i) local maps are represented as augmented keyframes instead of locally consistent submaps, (ii) the map building occurs online as opposed to offline, (iii) VO is performed in image space as opposed to Euclidean space, (iv) both VO and localizing against the map are performed in the same control cycle, and (v) the entire system is sensor generic (i.e., it can function with a stereo camera, lidar, or kinect sensor [4]).

2.2 Lidar-Based Localization

Laser scanners are used extensively for a number of scientific activities, such as archaeological site surveying (Allen et al., 2004), forest monitoring (Haala et al., 2004), traffic construction analysis (Kretschmer et al., 2004), medical applications (Antoine Maintz and Vierger, 1997), aerial topographic mapping (Brock et al., 2002), and robotic mapping and localization tasks (Borrmann et al., 2008). In most applications that use laser scanners, the goal is to align point cloud data, referred to as *scan matching* or *scan registration*, in order to build an accurate model or map of an environment. According to Zitova and Flusser (2003), there exist many different scan matching methods that vary in how they accomplish feature detection, feature matching, and image resampling and transformation. Perhaps the most common method for scan matching is the iterative closest point (ICP) algorithm (Besl and McKay, 1992), which uses the nearest neighbour assumption to establish point correspondences and minimizes a sum-of-squared-error objective function to compute the point cloud transformation. Other methods have used geometric primitives for point correspondence, such as surface normal vectors or local curvature of the scan points (Bae and Lichti, 2008), or search for similar geometric shapes, such as cylinders and planes (Bosse and Zlot, 2009; Rabbani and van den Heuvel, 2005). In addition to range data, laser scanners also provide intensity information, which has proven useful for the application of various feature detector/descriptors to accomplish data association.

[4] http://www.xbox.com/en-ca/kinect

Recognizing that laser intensity images[5] provide a gray scale image of a scene is not a new idea. Kretschmer et al. (2004) point out that in surveying, the reflectance images are often used by the surveyor to obtain a photo-realistic impression of the scanned area. In fact, most commercial surveyors use various reflective markers in the scene to act as tie points between different scan positions to make the data association problem much easier (Dold and Brenner, 2006). Bohm and Becker (2007) developed an automated marker-free method for point cloud registration that used point correspondences from the intensity images to estimate the rigid body transformations. SIFT features were extracted from the intensity images and Random Sample and Consensus (RANSAC) (Fischler and Bolles, 1981) was used for outlier detection. In order to dampen the areas of low and high reflectance, histogram equalization was used on all of the raw intensity images. Abymar et al. (2007) developed a technique to fuse laser data with digital camera images to generate coloured 3D point clouds. SIFT features were used to obtain correspondences between laser and the camera intensity images and RANSAC was used for outlier detection. They compared their automatic marker-free approach with a reflective marker approach and showed that the errors were of the same order of magnitude.

In the mobile robotics literature, few have actually used intensity information from a laser sensor for motion estimation. Neira et al. (1999) developed a sensor fusion technique in planar environments using their variant of the Extended Kalman Filter (EKF), called the SPfilter, which incorporated both range and intensity data from a laser scanner to localize against a known map. Guivant et al. (2000) described a SLAM system that used the intensity data from their laser scanner to identify reflective markers on landmarks in the environment, which simplified the data association problem. A similar strategy was used in the DARPA Urban Challenge, as several groups used intensity information from their laser sensors to detect lane markers and other cars (Urmson et al., 2008; Montemerlo et al., 2008). Yoshitaka et al. (2006b,a) developed what they call "Intensity-ICP", which uses the laser intensity data to help establish point correspondences for their ICP algorithm. Similar to classical ICP, point correspondences

[5] Also referred to as *reflectance images*.

are still established using the nearest neighbour method, but in addition to Euclidean distance, they also try to find points that have similar intensity values.

Although the above mentioned research used laser intensity information to aid in data association, they did not render the intensity data into an image to use local feature-based methods, making them very different from our work. To this author's knowledge, only two other research groups have used laser intensity images for motion estimation. May et al. (2009), and later Ye and Bruch (2010), developed 3D mapping and ego-motion estimation techniques using a Swiss Ranger Time of Flight (TOF) camera. Unlike a laser scanner, the Swiss Ranger uses an array of 24 LEDs to simultaneously illuminate a scene, offering the advantage of higher framerates. However, TOF cameras often have a limited FOV, short maximum range, and are very sensitive to environmental noise (May et al., 2009). Weingarten et al. (2004) were actually the first to use a TOF camera for robotics applications; however, their method, as well as others that followed (Droeschel et al., 2010; Yuan et al., 2009), only used range data from the sensor and not the intensity data. In contrast, May et al. (2009) used laser intensity images to employ two feature-based methods for motion estimation: a KLT-tracker and frame-to-frame VO using SIFT features[6]. Their results indicated that the SIFT approach yielded more accurate motion estimates than the KLT approach, but less accurate than their ICP method, which used a network-based global relaxation algorithm. Although May et al. demonstrated that frame-to-frame VO might be possible with a high-framerate TOF camera, the largest environment in which they tested was a 20m long indoor hallway, with no groundtruth. Furthermore, laser scanners are very different then TOF cameras in that they scan the scene with a single light source, introducing new problems such as image formation and image distortion caused by moving and scanning at the same time. McManus et al. (2011) were the first to use a laser scanner for appearance-based VO, and demonstrated two important results: (i) that 2D interest points are stable in lidar intensity images over drastic changes in ambient lighting and (ii) that stop-and-go lidar-based VO is comparable to stereo VO.

[6] Ye and Bruch (2010) presented a very similar SIFT approach using the Swiss Ranger.

Chapter 3

Image Formation

This chapter describes the image formation process where raw laser data is converted into a stack of azimuth, elevation, range, and intensity images. The intensity images are then enhanced for use in a keypoint detector/descriptor, which is a GPU implementation of the SURF (Bay et al., 2008) algorithm[1].

The first step in image formation is to develop a camera model, which requires knowledge of the specific sensor being used. The lidar used in this thesis is called the *Autonosys* and provides approximately equally spaced azimuth and elevation samples, making a spherical camera model a natural choice (more detail on this sensor is provided in Section 7). Figure 3.1 shows examples of a raw lidar intensity image and camera image of the scene for comparison. As is immediately evident, the raw lidar intensity image requires preprocessing in order to equalize the areas of high and low reflectance. The approach by McManus et al. (2011) preprocessed raw lidar images by applying adaptive histogram equalization and a Gaussian low-pass filter. Although this technique was shown to be successful, it fails to take into account the relationship between intensity and range (i.e., it does not adjust the intensity values based on the range measurements). Applying squared range corrections to the raw intensity images seems quite common in the literature (Donoghue et al., 2007; Holfe and Pfeifer, 2007; Luzum et al., 2004);

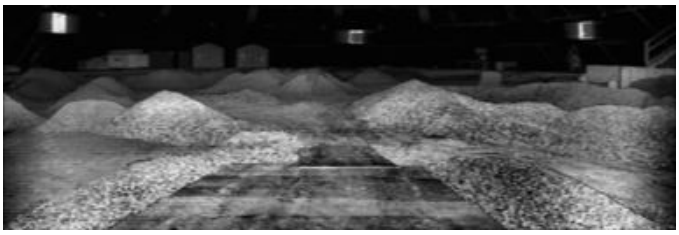
[1] http://asrl.utias.utoronto.ca/code/gpusurf/index.html

however, we found that applying a squared range correction darkened the image in the near field, which is not ideal for VO, since most of the tracked features are on the ground. Instead, we found that a linear range correction (i.e., multiplying the intensity values by their associated range) and rescaling the brightness values into the $[0, 255]$ range proved to work well (distant features become more visible, which are useful for orientation information). Figure 3.1 shows the processed intensity image using both adaptive histogram equalization and a linear range correction for comparison. To reiterate, a linear range correction was used for the 11km autonomous VT&R runs shown later in this thesis.

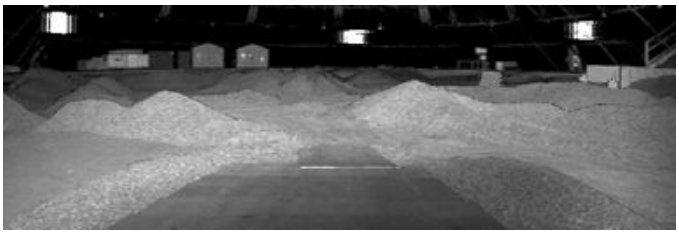
After processing the intensity image, the associated azimuth, elevation, and range data is assembled into an array in the exact same order as the intensity image, forming an *image stack*. This concept will prove useful in the next section, which introduces how keypoint measurements and their associated uncertainties are generated.

(a) Camera intensity image.

(b) Raw Autonosys intensity image.



(c) Processed Autonosys intensity image applying adaptive histogram equalization and a Gaussian low-pass filter.



(d) Processed Autonosys intensity image with a linear range correction. Note that distant features are more visible.

Figure 3.1: Illustrating the image processing stages required to transform raw intensity images into textured gray scale images. Image sizes have been adjusted to correspond to their field of view. Camera intensity image, 52.5^oV $\times$ 70^oH field of view (FOV), 512×384 pixels, 15Hz framerate. Autonosys intensity image, 30^oV $\times$ 90^oH FOV, 480×360 pixels, 2Hz framerate.

Chapter 4

Keypoint Generation

This chapter describes how keypoint measurements and associated uncertainties are generated using the image stack composed of azimuth, elevation, range, and intensity data. This measurement information is used both in the keypoint matching and outlier rejection step and forms the errors terms used in bundle adjustment.

4.1 Forming Measurements

The output of image formation is a stack of images, $\mathcal{I}$—intensity ($\mathcal{I}_\ell$), azimuth ($\mathcal{I}_\theta$), elevation ($\mathcal{I}_\phi$), and range ($\mathcal{I}_r$)—derived from the raw lidar output and shown in Figure 4.1. The stack may be evaluated at any integer row, r, and column, c, as, $\boldsymbol{\mathcal{I}}_{rc}$, a 4×1 column,

$$\boldsymbol{\mathcal{I}}_{rc} := \mathcal{I}(r, c) = \begin{bmatrix} \ell_{rc} & \theta_{rc} & \phi_{rc} & r_{rc} \end{bmatrix}^T,$$

where ℓ_{rc}, θ_{rc}, ϕ_{rc}, and r_{rc} are the scalar intensity, azimuth, elevation, and range stored at this location in the image stack. We assume that the elements of each image are independent, identically-distributed samples such that

$$\boldsymbol{\mathcal{I}}_{rc} = \overline{\boldsymbol{\mathcal{I}}}_{rc} + \delta\boldsymbol{\mathcal{I}}_{rc}, \qquad \delta\boldsymbol{\mathcal{I}}_{rc} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}\right), \qquad \mathbf{R} := \text{diag}\left\{\sigma_\ell^2, \sigma_\theta^2, \sigma_\phi^2, \sigma_r^2\right\},$$

where $\bar{\boldsymbol{\mathcal{I}}}_{rc}$ is the true value, $\delta\boldsymbol{\mathcal{I}}_{rc}$ is zero-mean Gaussian noise, and the components of $\mathbf{R}$ are based on the properties of the sensor (e.g., taken from the datasheet).

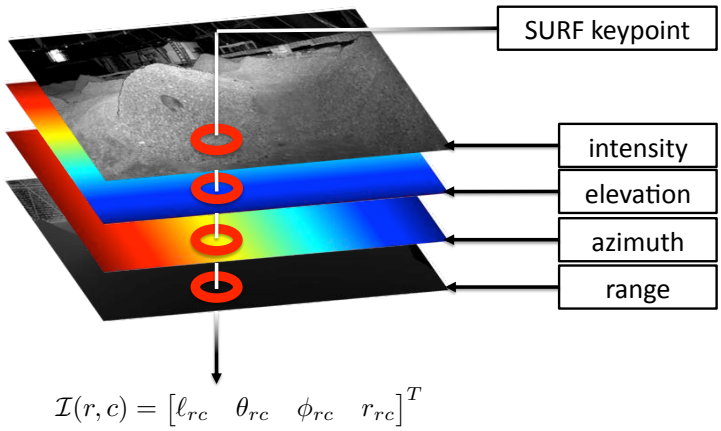


Figure 4.1: The image stack generated from the raw laser-rangefinder data. SURF keypoints are found in image space at sub-pixel locations and bilinear interpolation is used to find the azimuth, elevation, and range of the keypoint. Linearized error propagation from image space to azimuth/elevation/range is then used to determine the uncertainty of the measurement.

Keypoint detection returns a list of image locations, $\mathbf{y}_i = \begin{bmatrix} u & v \end{bmatrix}^T$, with associated covariances, $\mathbf{Y}_i$, where u, and v are generally not integers. We use bilinear interpolation of $\mathcal{I}$ to produce an azimuth/elevation/range measurement, $\mathbf{z}_i$. Given a pixel location, $\mathbf{y}_i$, in the neighbourhood of a set of four points that are arranged clockwise in a 2×2 patch, $\boldsymbol{S}_i = \{s_1, s_2, s_3, s_4\}$, the bilinear interpolation of that point is given by

$$\mathcal{B}(\mathbf{y}_i, \mathbf{S}_i) := (1 - du)(1 - dv)s_1 + du(1 - dv)s_2 + du\, dv\, s_3 + dv(1 - du)s_4,$$

where,

$$du := u - \lfloor u \rfloor, \qquad dv := v - \lfloor v \rfloor,$$

and $\lfloor \cdot \rfloor$ is the floor operator. Thus, to compute an interpolated azimuth, elevation, and range value, $\mathbf{z}_i$, a set of neighbouring points in the azimuth, elevation, and range images, $\boldsymbol{\mathcal{S}} := \{\boldsymbol{\mathcal{S}}_\theta, \boldsymbol{\mathcal{S}}_\phi, \boldsymbol{\mathcal{S}}_r\}$, are passed into a bilinear function, $\boldsymbol{\mathcal{B}}(\cdot)$, to produce a measurement according to

$$\mathbf{z}_i = \boldsymbol{\mathcal{B}}(\mathbf{y}_i, \boldsymbol{\mathcal{S}}) = \begin{bmatrix} \mathcal{B}(\mathbf{y}_i, \boldsymbol{\mathcal{S}}_\theta) & \mathcal{B}(\mathbf{y}_i, \boldsymbol{\mathcal{S}}_\phi) & \mathcal{B}(\mathbf{y}_i, \boldsymbol{\mathcal{S}}_r) \end{bmatrix}^T.$$

The uncertainty, $\mathbf{Q}_i$, associated with $\mathbf{z}_i$, is produced by propagation of $\mathbf{R}$ and $\mathbf{Y}_i$ through the

interpolation equations, such that $\mathbf{Q}_i := \mathbf{J}_i \mathbf{Y}_i \mathbf{J}_i^T + \mathbf{R}$, where $\mathbf{J}_i = \partial \boldsymbol{\mathcal{B}} / \partial \mathbf{y}|_{\mathbf{y}_i}$.

One of the challenges with using laser scanners in non-convex environments is that slight changes in sensor orientation can result in large range deviations depending on the geometry of the scene (e.g., objects that have a high angle of incidence with respect to the laser beam or thin objects can display large range deviations). To illustrate this, Figure 4.2 shows a standard-deviation range image, where each pixel represents the standard deviation of range values within a 3×3 local patch. As can be seen, structure boundaries, such as the hills, display the largest range deviations.

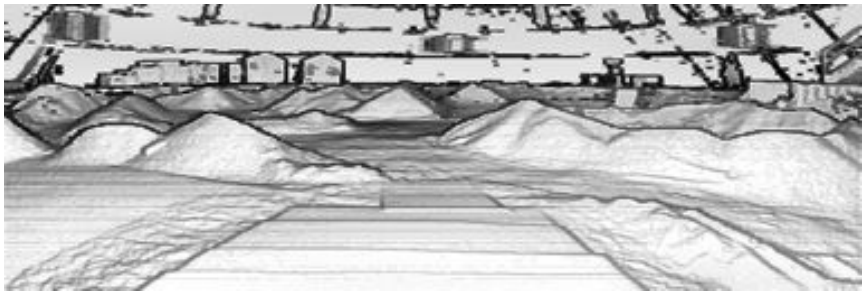


Figure 4.2: Standard deviation range image, where black represents the largest deviation. For visual clarity, adaptive histogram equalization has been applied to the raw standard deviation image. It is interesting to note that the standard deviation of neighbouring range values grows with distance along the ground plane as the spacing between points increases in the vertical direction. The maximum standard deviation was 26m.

4.2 Keypoint Matching

Once keypoint measurements have been generated, the following procedure is used to determine if two keypoints (each from a different image) could be possible matches. Instead of finding matches in image space, we find matches in measurement space, by looking for the nearest neighbour in range, azimuth, and elevation. This approach was taken because occasionally during data collection we lost data packets (i.e., we lost parts of an image), so matching within

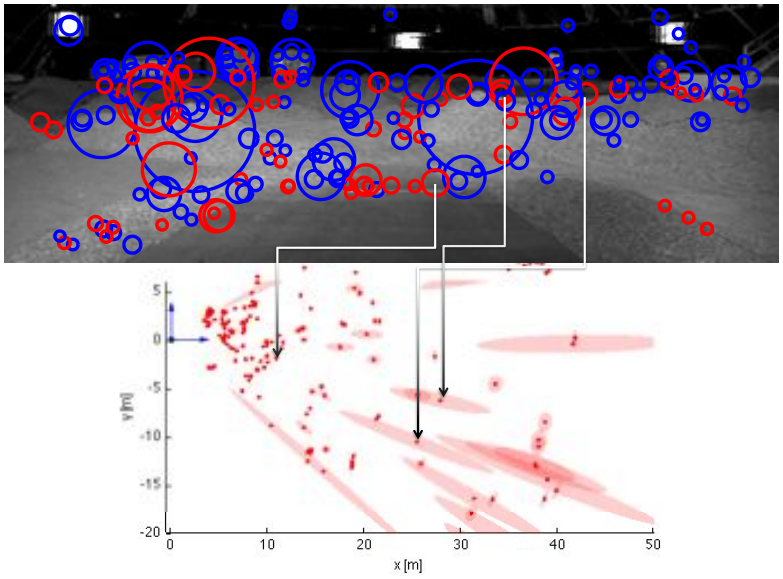


Figure 4.3: Top image: SURF keypoints detected in a preprocessed Autonosys image. Bottom image: the triangulated landmarks of each keypoint with the associated 3σ uncertainty ellipse shown in light red. Any keypoints that have large range deviations between neighbouring pixels will generate a large range uncertainty due to our linearized error propagation method. Thus, keypoints at structure boundaries, such as the reflective markers near the top of the image, will display large range uncertainty while points near the ground will generally have less range uncertainty.

a window in image space would not always find the appropriate nearest neighbours. It is important to note that since bilinear interpolation is used to compute keypoint measurements, any measurements with large range deviations (e.g., at structure boundaries) are discarded. This is because these measurements do not fit the smooth, linear model that is implicitly assumed when applying bilinear interpolation (see Figure 4.3 for an illustration of the different variance profiles for these landmarks).

Defining the measurement of keypoint i in image n as $\mathbf{z}_{n,i}$, and the measurement of keypoint j in image m as $\mathbf{z}_{m,j}$, the keypoint selection criteria are:

1. Keypoint range values must match to within some tolerance: $|r_{n,j} - r_{m,j}| < \delta_r$,

2. Keypoint azimuth/elevation angles must be located within a local neighbourhood: $\left\| \begin{bmatrix} \theta_{n,i} \\ \phi_{n,i} \end{bmatrix} - \begin{bmatrix} \theta_{m,j} \\ \phi_{m,j} \end{bmatrix} \right\| < \delta_{\theta,\phi}$,

3. The 64-element SURF descriptors must match to within some tolerance: $1 - \mathbf{d}_{n,i}{}^T \mathbf{d}_{m,j} < \delta_d$.

If there are multiple candidate matches for a particular keypoint, the candidate with the closest SURF descriptor is chosen as the possible match.

4.3 Outlier Rejection

RANSAC was used for outlier rejection and Horn's 3-point method (Horn, 1987) was used for hypothesis generation. To account for measurement uncertainties and for robustness, the well-known Geman-McClure estimator (Geman and McClure, 1987) is used along with Mahalonobis error metrics to compute the overall cost of each hypothesis. Recalling that each keypoint measurement, $\mathbf{z}_{k,j}$, corresponds to an observation of landmark j at time k, the error term, $\mathbf{e}_{k,j}$, is simply

$$\mathbf{e}_{k,j} := \mathbf{z}_{k,j} - \mathbf{g}\left(\mathbf{x}_{k-1,k}, \mathbf{p}_{k-1}^{j,k-1}\right),$$

where $\mathbf{x}_{k-1,k}$ is a 6×1 parameterization of $\mathbf{T}_{k-1,k}$, which is the 4×4 transformation from frame k to frame $k-1$, $\mathbf{p}_{k-1}^{j,k-1}$ is the vector from frame $k-1$ to landmark j expressed in frame $k-1$, and $\mathbf{g}(\cdot)$ builds the predicted azimuth/elevation/range value based on the current state estimate. Using the associated measurement uncertainty, $\mathbf{Q}_{k,j}$, the cost for each hypothesis is given by

$$\mathbf{E}_k := \sum_j w_j \mathbf{e}_{k,j}{}^T \mathbf{Q}_{k,j}^{-1} \mathbf{e}_{k,j}, \qquad w_j := \frac{1}{\left(\mathbf{e}_{k,j}{}^T \mathbf{Q}_{k,j}^{-1} \mathbf{e}_{k,j} + \sigma\right)},$$

where σ is an M-estimator parameter. After the best hypothesis is obtained, any matches with a Mahalanobis distance above a certain threshold are rejected; i.e., reject if $\mathbf{e}_{k,j}{}^T \mathbf{Q}_{k,j}^{-1} \mathbf{e}_{k,j} > e_{\max}$.

Chapter 5

Bundle Adjustment

This chapter deals with the core estimation theory behind the VT&R system. The central method that will be covered is *bundle adjustment* (Brown, 1958), which is a batch method that attempts to solve for the landmark positions and robot poses simultaneously, given the complete history of sensor measurements made by the robot. Bundle adjustment has quickly emerged as the dominant technique to filtering (Strasdat et al., 2010) and is used pervasively in Visual SLAM research (Konolige et al., 2010; Sibley et al., 2010).

Bundle adjustment is an optimization problem, where we seek to find the optimal state of robot poses, $\mathbf{x}$, and landmark positions, $\mathbf{p}$, that minimizes a weighted least-squares objective function. For the purposes of this derivation, we will assume that the state, $\mathbf{x}$, is a 6×1 parameterization of a 4×4 transformation matrix $\mathbf{T}$, and the landmarks, $\mathbf{p}$, are a 3×1 column of Euclidean points (i.e., we assume that the state and landmarks are members of a vector space). The error term we define is based on the difference between the observed keypoint location and the predicted keypoint location — referred to as *reprojection error*. Each measurement, $\mathbf{z}_{k,j}$, corresponds to an observation of landmark j at time k. Recalling from last section, the error term, $\mathbf{e}_{k,j}$, is given by

$$\mathbf{e}_{k,j} := \mathbf{z}_{k,j} - \mathbf{g}\left(\mathbf{x}_{0,k}, \mathbf{p}_0^{j,0}\right).$$

where $\mathbf{x}_{0,k}$ is a 6×1 parameterization of $\mathbf{T}_{0,k}$, which is the 4×4 transformation from frame k to

the base frame, $\mathbf{p}_0^{j,0}$ is the vector from the base frame to landmark j and expressed in the base frame, and $\mathbf{g}(\cdot)$ is the sensor model that computes a predicted azimuth/elevation/range value based on the current state estimate. We also assume that each measurement is corrupted by additive, zero-mean Gaussian noise, $\mathbf{n}_{k,j} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k,j})$.

Given K poses and M measurements, we can write our system in matrix form as

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_{1,1} \\ \vdots \\ \mathbf{z}_{K,1} \\ \mathbf{z}_{1,2} \\ \vdots \\ \mathbf{z}_{K,M} \end{bmatrix}, \quad \mathbf{x} := \begin{bmatrix} \mathbf{x}_{0,1} \\ \vdots \\ \mathbf{x}_{0,K} \end{bmatrix}, \quad \mathbf{p} := \begin{bmatrix} \mathbf{p}_0^{1,0} \\ \vdots \\ \mathbf{p}_0^{M,0} \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}, \mathbf{p}) := \begin{bmatrix} \mathbf{g}\left(\mathbf{x}_{0,1}, \mathbf{p}_0^{1,0}\right) \\ \vdots \\ \mathbf{g}\left(\mathbf{x}_{0,K}, \mathbf{p}_0^{1,0}\right) \\ \mathbf{g}\left(\mathbf{x}_{0,1}, \mathbf{p}_0^{2,0}\right) \\ \vdots \\ \mathbf{g}\left(\mathbf{x}_{0,K}, \mathbf{p}_0^{M,0}\right) \end{bmatrix} \tag{5.1}$$

$$\mathbf{Q} := \text{diag}\left(w_{1,1}\mathbf{Q}_{1,1}, \ldots, w_{K,1}\mathbf{Q}_{K,1}, w_{1,2}\mathbf{Q}_{1,2}, \ldots, w_{K,M}\mathbf{Q}_{K,M}\right), \tag{5.2}$$

where we have included the Geman-McClure estimator,

$$w_{i,j} := \frac{1}{\left(\mathbf{e}_{i,j}^T \mathbf{Q}_{i,j}^{-1} \mathbf{e}_{i,j} + \sigma\right)},$$

for robustness to outliers. Using the above quantities, we can then define the standard Mahalonobis objective function as

$$J(\mathbf{z}|\mathbf{x}, \mathbf{p}) := \frac{1}{2} \left(\mathbf{z} - \mathbf{g}(\mathbf{x}, \mathbf{p})\right)^T \mathbf{Q}^{-1} \left(\mathbf{z} - \mathbf{g}(\mathbf{x}, \mathbf{p})\right). \tag{5.3}$$

Since (5.3) is nonlinear in the design variables, we linearize via a first-order Taylor series expansion:

$$J(\mathbf{z}|\bar{\mathbf{x}}+\delta\mathbf{x}, \bar{\mathbf{p}}+\delta\mathbf{p}) \approx \frac{1}{2} \left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) + \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{p}\right)^T \mathbf{Q}^{-1} \left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) + \mathbf{A}\delta\mathbf{x} + \mathbf{B}\delta\mathbf{p}\right), \tag{5.4}$$

where $\mathbf{A} := -\partial\mathbf{g}/\partial\mathbf{x}$ and $\mathbf{B} := -\partial\mathbf{g}/\partial\mathbf{p}$. Taking the derivative of (5.4) with respect to the perturbations of the state variables, $\{\delta\mathbf{x}, \delta\mathbf{p}\}$, and setting $\partial J/\partial\{\delta\mathbf{x}, \delta\mathbf{p}\}$ to zero results in the

following system of equations

$$\begin{aligned}
\mathbf{0} &= \begin{bmatrix}\mathbf{A}^T \\ \mathbf{B}^T\end{bmatrix} \mathbf{Q}^{-1} \left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) + \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{p}\end{bmatrix} \right), \\
\implies \begin{bmatrix}\mathbf{A}^T \\ \mathbf{B}^T\end{bmatrix} \mathbf{Q}^{-1} \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{p}\end{bmatrix} &= - \begin{bmatrix}\mathbf{A}^T \\ \mathbf{B}^T\end{bmatrix} \mathbf{Q}^{-1} \left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \right), \\
\begin{bmatrix}\mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A} & \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{B} \\ \mathbf{B}^T\mathbf{Q}^{-1}\mathbf{A} & \mathbf{B}^T\mathbf{Q}^{-1}\mathbf{B}\end{bmatrix} \begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{p}\end{bmatrix} &= - \begin{bmatrix}\mathbf{A}^T\mathbf{Q}^{-1} \\ \mathbf{B}^T\mathbf{Q}^{-1}\end{bmatrix} \left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) \right). \qquad (5.5)
\end{aligned}$$

Interestingly, the coefficient matrix on the left side of the above system of equations[1] has a very specific sparsity pattern that can be exploited for a more efficient solution. To see how this sparsity pattern results, we turn our attention to the linearized error terms:

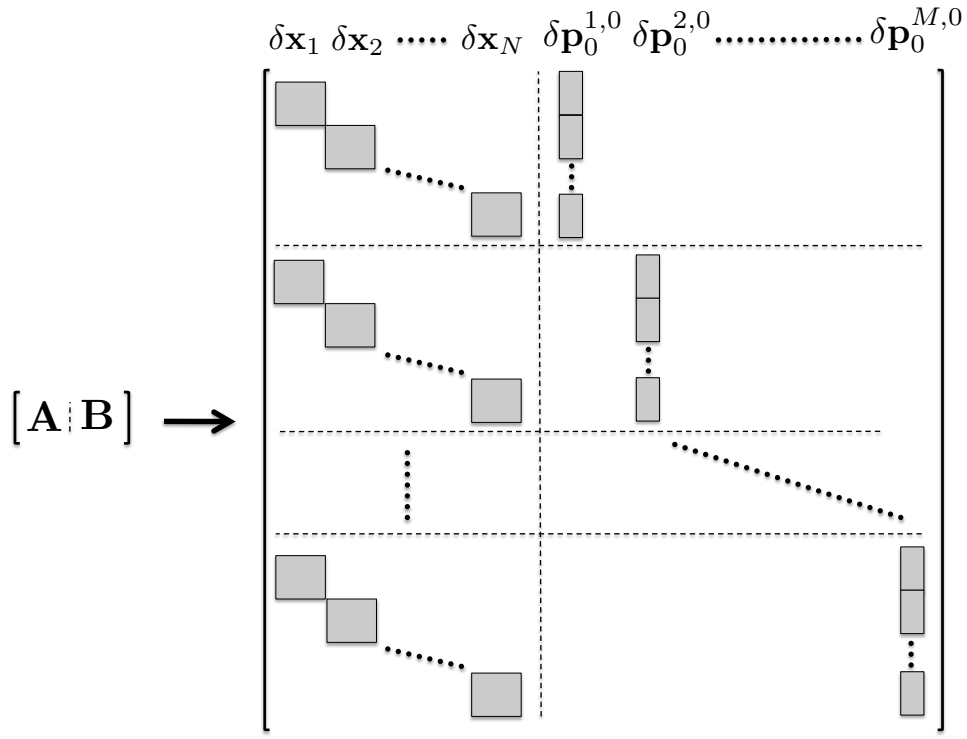
$$\mathbf{e}(\mathbf{z}|\bar{\mathbf{x}} + \delta\mathbf{x}, \bar{\mathbf{p}} + \delta\mathbf{p}) \approx \mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}}) + \begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix} \begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{p}\end{bmatrix}.$$

Noting how the state terms were arranged in (5.1), we see that the Jacobian matrices, $\mathbf{A}$, and $\mathbf{B}$, take the form illustrated in Figure 5.1(a). The resulting sparsity pattern of the coefficient matrix in equation (5.5) is shown in Figure 5.1(b).

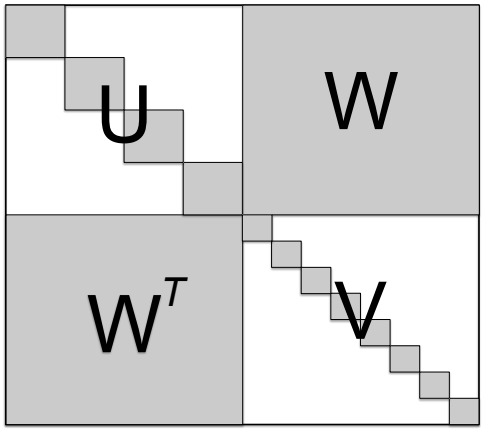
Recognizing the fact that in most robot localization problems the number of landmarks will be much larger than the number of robot poses (i.e., dim $\mathbf{p} \gg$ dim $\mathbf{x}$), one can marginalize[2] out the landmark positions in order to directly solve for the robot poses (Brown, 1958). This marginalization is accomplished through use of the Schur complement and leads to the much more efficient *sparse bundle adjustment* formulation. To show this, we rewrite the system of equations in (5.5) as

$$\begin{bmatrix}\mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V}\end{bmatrix} \begin{bmatrix}\delta\mathbf{x} \\ \delta\mathbf{p}\end{bmatrix} = - \begin{bmatrix}\mathbf{e}_x \\ \mathbf{e}_p\end{bmatrix}$$

[1] Called the *information matrix*, which is simply the inverse of the covariance matrix.
[2] This is the process of eliminating terms in the system of equations via elementary row operations.



(a) Sparsity pattern in Jacobian matrices.



(b) Sparsity pattern in the bundle adjustment coefficient matrix.

Figure 5.1: Sparsity patterns in the Jacobians and coefficient matrix, where shaded regions represent non-zero elements and white regions represent zeros.

where

$$\begin{aligned}
\mathbf{U} &:= \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{A}, \\
\mathbf{W} &:= \mathbf{A}^T\mathbf{Q}^{-1}\mathbf{B}, \\
\mathbf{V} &:= \mathbf{B}^T\mathbf{Q}^{-1}\mathbf{B}, \\
\mathbf{e}_x &:= \mathbf{A}^T\mathbf{Q}^{-1}\left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}})\right), \\
\mathbf{e}_p &:= \mathbf{B}^T\mathbf{Q}^{-1}\left(\mathbf{z} - \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{p}})\right).
\end{aligned}$$

Multiplying by the Schur complement results in

$$\begin{bmatrix} \mathbf{1} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = - \begin{bmatrix} \mathbf{1} & -\mathbf{W}\mathbf{V}^{-1} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{e}_x \\ \mathbf{e}_p \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T & \mathbf{0} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{p} \end{bmatrix} = - \begin{bmatrix} \mathbf{e}_x - \mathbf{W}\mathbf{V}^{-1}\mathbf{e}_p \\ \mathbf{e}_p \end{bmatrix}.$$

Since $\mathbf{V}$ is block-diagonal, its inverse is inexpensive to compute. After solving directly for $\delta\mathbf{x}$, $\delta\mathbf{p}$ can be computed via back substitution and the state variables are updated according to $\mathbf{x} \leftarrow \mathbf{x} + \delta\mathbf{x}$ and $\mathbf{p} \leftarrow \mathbf{p} + \delta\mathbf{p}$. This iterative process is continued until the norm of the perturbations is below a certain threshold.

However, as the above formulation is batch and intended for offline processing, a sliding window approach is used for online estimation. The window at timestep k includes two poses, $\mathbf{x}_{0,k-1}$ and $\mathbf{x}_{0,k}$, and all of the matching landmarks between the two frames (suppose there are M_k matches). However, only the current state, $\mathbf{x}_{0,k}$, and the M_k landmark positions, $\mathbf{p}_0^{j,0}$, are design variables in the optimization; the previous state, $\mathbf{x}_{0,k-1}$ is held fixed, but is still used in computing the error terms. In essence, this is a form of frame-to-frame VO that is based on sparse bundle adjustment. Although this work could be extended to optimize over multiple frames, we found it unnecessary for VT&R.

A no-motion prior on the pose of the vehicle at timestep k, denoted by the density $\{\mathbf{x}'_{0,k}, \mathbf{P}_k\}$, is also included in order to bound the estimate within a local neighbourhood of its previous location and prevent any spurious estimates. To add a no-motion prior, we simply include an

extra term in the objective function given by equation (5.3):

$$J(\mathbf{z}_k, |\mathbf{x}_k, \mathbf{p}_k) := \frac{1}{2} \begin{bmatrix} \mathbf{x}'_{0,k} - \mathbf{x}_{0,k} \\ \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) \end{bmatrix}^T \begin{bmatrix} \mathbf{P}_k^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{0,k} - \mathbf{x}_{0,k} \\ \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) \end{bmatrix},$$

where

$$\mathbf{z}_k := \begin{bmatrix} \mathbf{z}_{k-1,j} \\ \mathbf{z}_{k,j} \\ \vdots \\ \mathbf{z}_{k-1,j+M_k} \\ \mathbf{z}_{k,j+M_k} \end{bmatrix}, \quad \mathbf{x}_k := \begin{bmatrix} \mathbf{x}_{0,k-1} \\ \mathbf{x}_{0,k} \end{bmatrix}, \quad \mathbf{p}_k := \begin{bmatrix} \mathbf{p}_0^{j,0} \\ \vdots \\ \mathbf{p}_0^{j+M_k,0} \end{bmatrix},$$

$$\mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) := \begin{bmatrix} \mathbf{g}(\mathbf{x}_{0,k-1}, \mathbf{p}_0^{j,0}) \\ \mathbf{g}(\mathbf{x}_{0,k}, \mathbf{p}_0^{j,0}) \\ \vdots \\ \mathbf{g}(\mathbf{x}_{0,k-1}, \mathbf{p}_0^{j+M_k,0}) \\ \mathbf{g}(\mathbf{x}_{0,k}, \mathbf{p}_0^{j+M_k,0}) \end{bmatrix}, \quad \mathbf{Q}_k := \text{diag}\left(\mathbf{Q}_{k-1,j}, \mathbf{Q}_{k,j}, \ldots, \mathbf{Q}_{k-1,j+M_k}, \mathbf{Q}_{k,j+M_k}\right),$$

and we have introduced the timestep k to denote that this is a sliding window approach. Linearizing the system in a similar manner as before yields the following system of equations:

$$\begin{aligned}
\begin{bmatrix} \mathbf{C}_k^T & \mathbf{A}_k^T \\ \mathbf{0} & \mathbf{B}_k^T \end{bmatrix} \begin{bmatrix} \mathbf{P}_k^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{C}_k & \mathbf{0} \\ \mathbf{A}_k & \mathbf{B}_k \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{0,k} \\ \delta\mathbf{p}_k \end{bmatrix} &= - \begin{bmatrix} \mathbf{C}_k^T & \mathbf{A}_k^T \\ \mathbf{0} & \mathbf{B}_k^T \end{bmatrix} \begin{bmatrix} \mathbf{P}_k^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{0,k} - \mathbf{x}_{0,k} \\ \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) \end{bmatrix}, \\
\begin{bmatrix} \mathbf{C}_k^T\mathbf{P}_k^{-1}\mathbf{C}_k + \mathbf{A}_k^T\mathbf{Q}_k^{-1}\mathbf{A}_k & \mathbf{A}_k^T\mathbf{Q}_k^{-1}\mathbf{B}_k \\ \mathbf{B}_k^T\mathbf{Q}_k^{-1}\mathbf{A}_k & \mathbf{B}_k^T\mathbf{Q}_k^{-1} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{0,k} \\ \delta\mathbf{p}_k \end{bmatrix} &= - \begin{bmatrix} \mathbf{C}_k^T\mathbf{P}_k^{-1} & \mathbf{A}_k^T\mathbf{Q}_k^{-1} \\ \mathbf{0} & \mathbf{B}_k^T\mathbf{Q}_k^{-1}\mathbf{B}_k \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{0,k} - \mathbf{x}_{0,k} \\ \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) \end{bmatrix}, \\
\begin{bmatrix} \mathbf{P}_k^{-1} + \mathbf{A}_k^T\mathbf{Q}_k^{-1}\mathbf{A}_k & \mathbf{A}_k^T\mathbf{Q}_k^{-1}\mathbf{B}_k \\ \mathbf{B}_k^T\mathbf{Q}_k^{-1}\mathbf{A}_k & \mathbf{B}_k^T\mathbf{Q}_k^{-1} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{0,k} \\ \delta\mathbf{p}_k \end{bmatrix} &= - \begin{bmatrix} -\mathbf{P}_k^{-1} & \mathbf{A}_k^T\mathbf{Q}_k^{-1} \\ \mathbf{0} & \mathbf{B}_k^T\mathbf{Q}_k^{-1}\mathbf{B}_k \end{bmatrix} \begin{bmatrix} \mathbf{x}'_{0,k} - \mathbf{x}_{0,k} \\ \mathbf{z}_k - \mathbf{g}(\mathbf{x}_k, \mathbf{p}_k) \end{bmatrix}, \\
\begin{bmatrix} \mathbf{U}' & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{0,k} \\ \delta\mathbf{p}_k \end{bmatrix} &= - \begin{bmatrix} \mathbf{e}'_x \\ \mathbf{e}_p \end{bmatrix},
\end{aligned} \tag{5.6}$$

where $\mathbf{C}_k := -\frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_k} = -\mathbf{1}$.

As discussed earlier, using the Schur compliment, one can marginalize the landmarks onto the poses and efficiently solve for the pose variables, $\mathbf{x}_{0,k}$. The landmarks, $\mathbf{p}_0^{j,0}$ are computed via backsubstitution. The Levenburg-Marqaurdt (LM) algorithm (Levenberg, 1944) was chosen as the gradient-based optimization method due to its rich heritage in bundle adjustment problems. The LM method works by augmenting the coefficient matrix in equation (5.6) with a diagonal matrix $\lambda\mathbf{1}$, where $\lambda > 0$ and referred to as the *damping parameter*. Depending on the change in the objective function, this damping parameter is adjusted after each iteration and adapts the convergence properties of LM to be similar to either steepest-descent or the Gauss-Newton method. The rule for adapting λ is the following:

$$\lambda = \begin{cases} \beta\lambda, \text{where } \beta > 0 & \text{if } J_k - J_{k-1} > 0 \\ \eta\lambda, \text{where } \eta < 0 & \text{otherwise} \end{cases}$$

Chapter 6

System Overview

This chapter provides a detailed overview of the VT&R system, combining all the concepts and methods shown in earlier chapters. A description of the online mapping process during the teach pass will be covered, as well as the local map construction used during the repeat pass. In addition, a block diagram of the system architecture is provided, along with detailed descriptions of each major code block and how they interact with one another.

6.1 The Teach Pass

During the teach pass, the system builds a topologically connected network of keyframes, which is either added to or begins the creation of a *pose graph* (Sibley et al., 2010). For each keyframe, the following information is stored.

- **Keypoints and Descriptors** - A list of keypoints, $\mathbf{z}_i$, associated uncertainties, $\mathbf{Q}_i$, and associated 64-element SURF descriptors, as described in Section 4.
- **Camera Calibration/Geometry Information** - Sensor-specific camera geometry used to convert a keypoint, $\mathbf{z}_i$, to a Euclidean landmark, $\mathbf{p}_i$, and vice versa.
- **Timestamps** - Used for synchronization.
- **Images** - Used purely for visualization of feature tracks.

Frame-to-frame transformation matrices and their associated uncertainties are stored along edges that connect two keyframes in the pose graph, as well as a *matchlist*, which specifies the post-RANSAC matching keypoints between the two frames. This pose graph structure is illustrated in Figure 6.1.

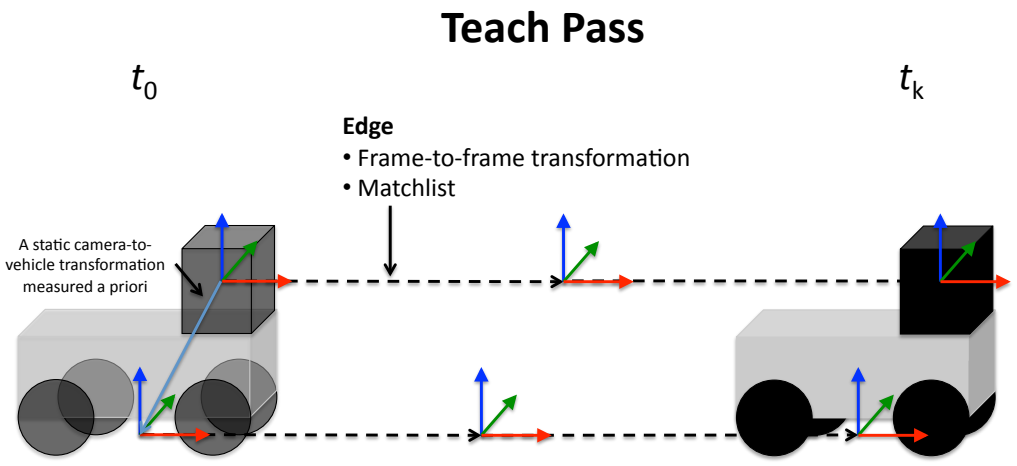


Figure 6.1: The taught path is built as a pose graph consisting of relative frame transformations between poses. Vertices in the graph store keyframes containing keypoints (e.g., azimuth, elevation, range), SURF descriptors, camera calibration/geometry information, and timestamps. Edges store relative frame transformations and lists of inter-frame keypoint matches. New vertices are added to the graph when the robot travels a certain distance or when it rotates by a certain amount. Once the taught path is constructed, the path is transformed into the vehicle reference frame using a camera-to-vehicle transformation.

6.2 The Repeat Pass

When repeating a route, the system acquires the appropriate chain of relative transformations from the pose graph (in the order that is specified) and constructs the taught route in the vehicle base frame. Since the route was constructed in the frame of the camera, we transform the path

into the vehicle base frame according to the following:

$$\mathbf{T}_{v_0,v_k} = \mathbf{T}_{v,c}\mathbf{T}_{c_0,c_k}\mathbf{T}_{v,c}^{-1},$$

where $\mathbf{T}_{v,c}$ is the camera-to-vehicle transformation, which is a fixed transformation that is measured a priori and $\mathbf{T}_{c_0,c_k}$ is the transformation from frame k to frame 0 as seen in the camera base frame for this particular path. Once the path has been built in the vehicle frame, at each timestep, the system performs the following steps for localization (see Figure 6.2).

1. **Frame-to-frame VO** - This provides an incremental pose update to achieve a good guess for the next step of localizing against the nearest keyframe. Keypoint matching and outlier rejection are accomplished using the techniques described in Section 4 and the bundle adjustment formulation in Section 5 is used for frame-to-frame VO. It should be noted that the estimation takes place in the nearest keyframe's reference frame, called the *branch*. This makes the approach completely relative, as the estimation is never performed in a fixed global reference frame.

2. **Localization against the map** - The system localizes against the nearest keyframe on the pose graph (nearest in a Euclidean sense), using the keypoint matching and outlier rejection methods in Section 4. This provides a relative transformation estimate, $\mathbf{T}_{c_b,c_k}$, between the current camera pose at time k, called the *leaf*, and the nearest keyframe, called the *branch*. As is done for frame-to-frame VO, the estimation is done in the branch reference frame. After matching against the map, the new estimate is transformed into the vehicle frame for the path tracker according to: $\mathbf{T}_{v_0,v_k} = \mathbf{T}_{v,c}\mathbf{T}_{c_0,c_b}\mathbf{T}_{c_b,c_k}\mathbf{T}_{c,v}$, where $\mathbf{T}_{v,c}$ is the fixed camera-to-vehicle transformation, $\mathbf{T}_{c_0,c_b}$ is the transformation from the current branch to the camera base frame, and $\mathbf{T}_{c_b,c_k}$ is the newly updated leaf-to-branch transformation.

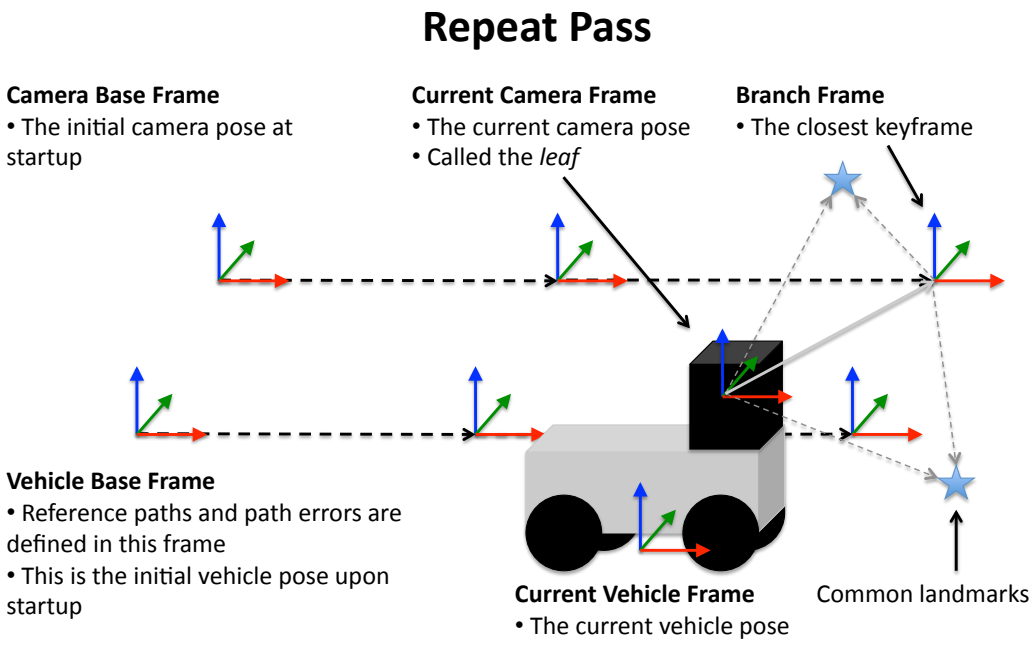


Figure 6.2: During the repeat pass, images from the current sensor frame, called the *leaf*, are used for a frame-to-frame VO estimate and then matched against the nearest keyframe from the teach pass, called the *branch*.

6.2.1 The Sliding Local Map

This keyframe-to-keyframe matching is clearly less costly than a multi-frame bundle adjustment method, but it does give up accuracy to a multi-frame approach because it only considers the nearest keyframe. During preliminary testing, it was discovered that simple keyframe-to-keyframe matching was not robust enough to large movements and the algorithm would often fail to localize against the map. Inspired by the continuous relative representation of Sibley et al. (2010), we addressed this problem by introducing a *sliding local map*, which attempts to embed the nearest keyframe with additional information from the surrounding keyframes (i.e., we augmented the closest keyframe with surrounding keypoints). This is accomplished in the following way (see Figure 6.3 for an illustration).

1. Pick a window of keyframes surrounding the nearest keyframe at timestep k.

2. For each common landmark between keyframes $k-i$ and $k-i+1$ in the set of all matches,

M_i, compute the keypoint measurement in the branch keyframe using the sensor model:

$$\forall m \in M_i, \text{compute } \mathbf{z}_{k,m} = \mathbf{g}\left(\mathbf{x}_{k-i+1,k}, \mathbf{p}_{k-i+1}^{m,k-i+1}\right),$$

where we note that the relative transformations from $k-i+1$ to k are all available from the pose graph. These additional keypoints are added to the branch in order to include additional information in the local map.

3. After the local map has been built up from all of the surrounding keyframes, finding keypoint matches, rejecting outliers, and computing the corrected pose of the vehicle follows the exact same procedure as outlined in previous chapters.

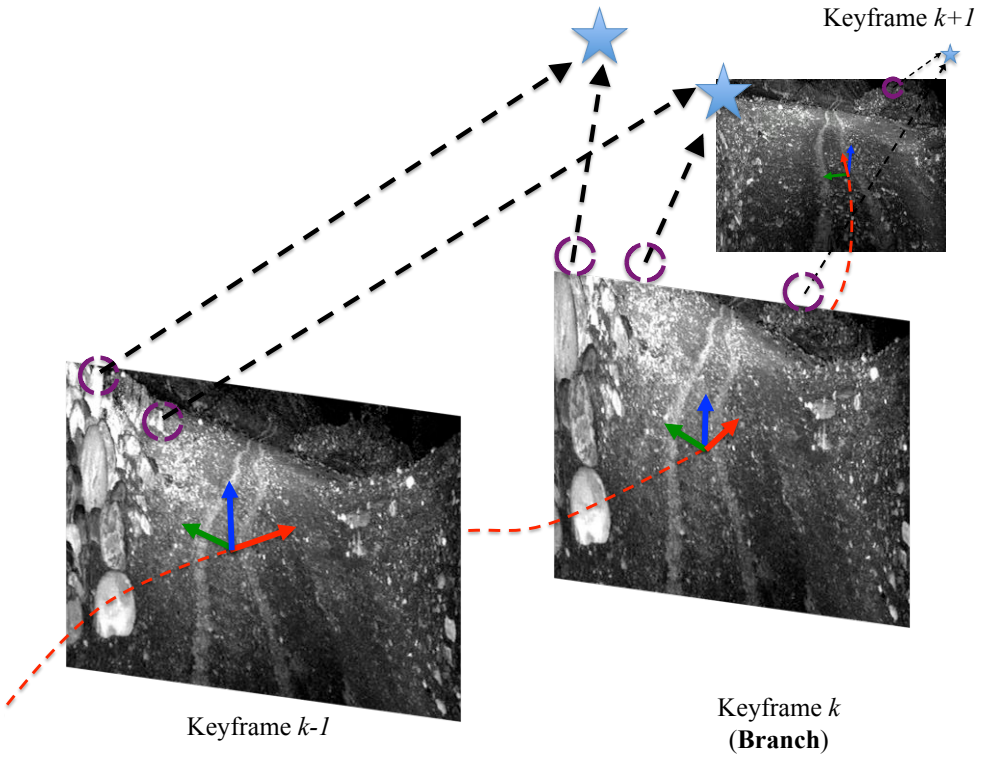


Figure 6.3: An illustration of the local map construction, where the nearest keyframe, called the *branch*, is embedded with keypoints from surrounding keyframes. Including additional keypoints in this manner increases map matching due to the non-identical teach and repeat trajectories that lead to slight viewpoint changes.

Table 6.1: Repeat pass failure mode parameters

Parameter	Description	Value
τ	maximum distance without localizing against the map	3m
N_k	number of keypoint matches for successful match	10
N_m	number of consecutive localizations required	5

6.3 Handling Off-Nominal Cases

There are numerous off-nominal modes that can occur while repeating the route, which include failing to localize against the map and/or frame-to-frame VO failures. These can occur when there is sufficient motion blur, large path deviations leading to viewpoint changes, scene changes (e.g., due to rain moistening the ground), or lost image packets. In order to be robust to such failure modes, the system responds in the following ways. If localizing against the map is unsuccessful, the system will continue to move and use frame-to-frame VO up to a specific distance, τ. Afterwards, if still unable to match against the map, the vehicle will stop and enter a search mode where the current image is matched against a series of images in the database around the latest branch estimate. This search mode will continue until it exhaustively searches all the images in the database. A successful match against the map occurs if more than N_k keypoints are matched N_m times. Once a successful match has been determined, the localization estimate is updated and the vehicle continues to follow the path.

If frame-to-frame VO fails, but the system is still able to localize against the map, the vehicle will continue to follow the path indefinitely. If both frame-to-frame VO and matching against the map fails, then the vehicle stops and enters its search mode. The parameters used in these experiments are provided in Table 6.1.

6.4 System Architecture

The VT&R system was developed using Robot Operating System (ROS)[1], which is open-source middleware designed specifically for robotics applications. The system architecture is illustrated in Figure 6.4, which highlights all of the major code blocks and how they interact with one another. The entire framework was designed to be sensor-generic, meaning that it can work with either stereo camera input or lidar input and can be generalized to other sensors in the future[2].

The specific code blocks developed as a part of this thesis are the following: (i) the Autonosys lidar driver, (ii) the Autonosys keypoint geometry, (iii) building the sliding local map, and (iv) the path tracker. A more detailed description of these code blocks is provided below.

6.4.1 Autonosys Lidar Driver

Raw data packets containing azimuth, elevation, range, and intensity are broadcast from a computer connected to the Autonosys lidar. As the software driver for the sensor is only available on Windows and the VT&R system is run on Linux, two separate computers were required. The Windows machine would capture the data and then port-forward the data packets to the Linux machine. Once the data packets are received in ROS, the Autonosys lidar Driver converts it into an image stack, as described in section 3.

6.4.2 Autonosys Keypoint Geometry and Keypoint Matching

These two code blocks are responsible for generating keypoint measurements from the image stack and finding candidate keypoint matches, as described in section 4. The keypoint geometry code block transforms keypoints to homogeneous/non-homogeneous Euclidean points and vice versa. Referring to the keypoint matching criteria outlined in section 4, keypoint matches are

[1] http://www.ROS.org

[2] It should be noted that Paul Furgale was the principal software developer who designed this sensor-generic framework.

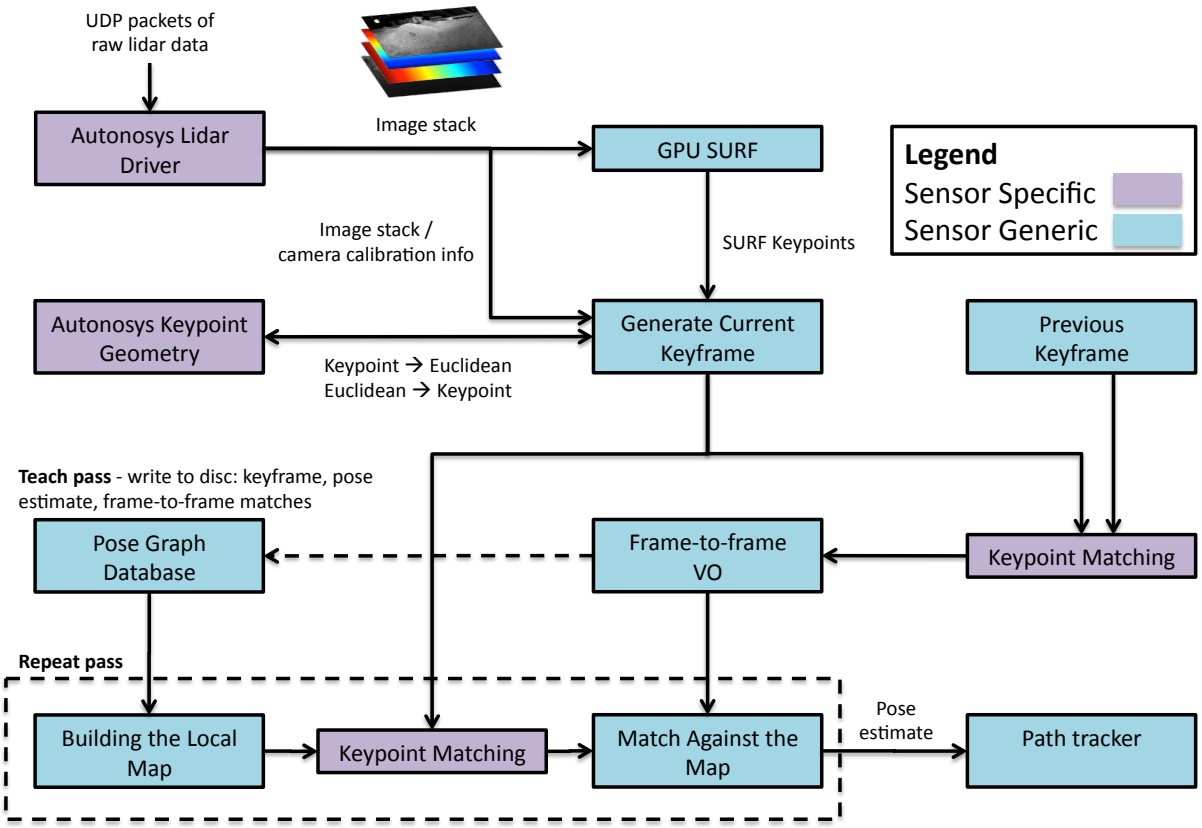


Figure 6.4: Detailed VT&R system architecture. Note the difference between the sensor specific and sensor generic components of the system.

rejected if their range deviation exceeds 5m or if the norm of their azimuth/elevation angles are greater than 30% of the maximum horizontal FOV / vertical FOV, respectively.

6.4.3 Sliding Local Map Implementation

For online performance, it was necessary to ignore duplicate landmarks when building the local map. This is relatively straightforward bookkepping procedure, as the pose graph contains the match lists between adjacent keyframes, which is an $N \times 2$ matrix where each row, $[n, m]$, represents a match between keypoint n from keyframe k and keypoint m from keyframe $k+1$. By using these match lists and assigning each keypoint a unique identifier, duplicate keypoints can easily be recognized and ignored.

Choosing an appropriate window size for the local map is an important consideration, be-

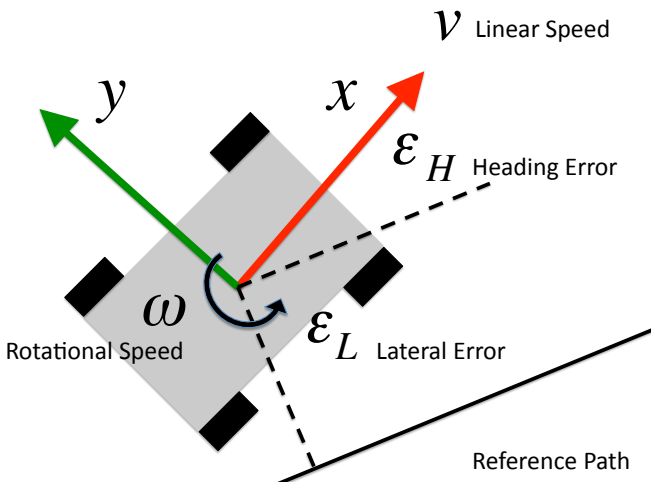


Figure 6.5: Path tracking errors.

cause a larger window results in a larger number of keypoints against which to match. There is of course a limit to how large the window should be, as it increases the computational burden to both construct and match against a larger map and it may also reduce the localization accuracy after a certain point, since larger maps accumulate more error than smaller maps. Furgale and Barfoot (2010) found that local maps on the order of 5m long were ideal for their system. Out of a concern for computational efficiency, a window size of approximately 2.5m was used in the system presented in this thesis; however, performing a more detailed analysis on the optimal map size is currently the focus of future work.

6.4.4 Path tracker

This controller is based on the path tracker described by Marshall et al. (2008) and later adapted by McManus (2009), which is a nonlinear controller that uses full-state feedback linearization. As it was originally designed for planar environments, path errors are computed in the local vehicle reference frame, in order to track 3D paths. Several other engineering heuristics are also applied for optimal performance, such as measuring the heading error in front of the vehicle for smoother turns.

This path tracking controller makes the following assumptions: (i) the linear speed of the

vehicle is a constant, (ii) the vehicle kinematics can be appropriately described by a unicycle model, and (iii) the reference path is linear. Under these assumptions, one can define the error quantities shown in Figure 6.5. The system of equations describing the lateral/heading error rates is given by

$$\begin{bmatrix} \dot{\varepsilon}_L \\ \dot{\varepsilon}_H \end{bmatrix} = \begin{bmatrix} v \sin \varepsilon_H \\ \omega \end{bmatrix},$$

where ε_H is the heading error, ε_L is the lateral error, v is the linear speed, which is assumed constant, and ω is the rotational control input. A substitution of variables is now introduced to transform the above nonlinear system into a linear system. Let $z_1 := \varepsilon_L$ and $z_2 := v \sin \varepsilon_H$. The new system of equations is given by

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ v \cos \varepsilon_H \omega \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \eta \end{bmatrix},$$

where a new quantity has been conveniently defined: $\eta := v \cos \varepsilon_H \omega$ (note that η is the new control input in these transformed coordinates). Choosing a proportional controller of the form $\eta = -k_1 z_1 - k_2 z_2$ gives the following closed-loop error dynamics:

$$\dot{\mathbf{z}} = \begin{bmatrix} 0 & 1 \\ -k_1 & k_2 \end{bmatrix} \mathbf{z}.$$

As long as $k_1, k_2 > 0$, this system will be stable (Marshall et al., 2008). Using the two definitions of η, one can solve for the control input ω, which is given by

$$\omega = \frac{-k_1 \varepsilon_L - k_2 v \sin \varepsilon_H}{v \cos \varepsilon_H}.$$

It is assumed that the inputed reference path is defined as a set of 3D poses, given by transformation matrices: i.e., $\mathcal{P} := \{\mathbf{T}_{0,1}, \ldots, \mathbf{T}_{0,N}\}$ and at each timestep, the path tracker receives the vehicle transformation matrix, $\mathbf{T}_{0,v}$. The path errors are computed relative to the local vehicle reference frame according to the follow procedure.

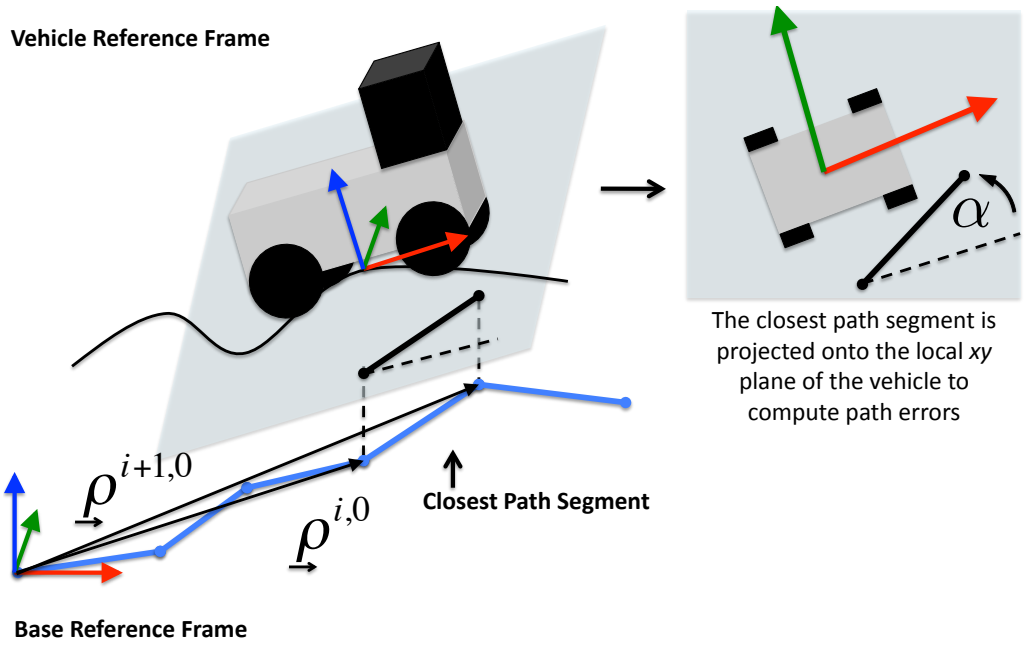


Figure 6.6: Coordinate frame definitions.

1. Compute position vectors to the closest two path points, expressed in the vehicle frame:

$$\begin{bmatrix} \mathbf{p}_v^{v,i} \\ 1 \end{bmatrix} = \mathbf{T}_{0,v}^{-1} \begin{bmatrix} \mathbf{p}_0^{0,i} \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} \mathbf{p}_v^{v,i+1} \\ 1 \end{bmatrix} = \mathbf{T}_{0,v}^{-1} \begin{bmatrix} \mathbf{p}_0^{0,i+1} \\ 1 \end{bmatrix}.$$

2. Compute the yaw angle that rotates the current vehicle frame in the same direction as the current path segment (see Figure 6.6):

$$\alpha := -\text{atan2}\left(\mathbf{p}_v^{v,i+1}(2) - \mathbf{p}_v^{v,i}(2), \mathbf{p}_v^{v,i+1}(1) - \mathbf{p}_v^{v,i}(1)\right),$$

where the notation $\mathbf{p}(n)$ means the n^{th} component of $\mathbf{p}$. Note that α is equal to ε_H as defined in Figure 6.5. However, for reasons that will be explained shortly, we took a different approach to measuring the heading error that is based on the orientations of the neighbouring keyframes.

3. Define a new coordinate frame that is contained in the vehicle's xy plane, and has its x-axis pointing along the line segment projected onto this xy plane (see Figure 6.7(a)). This defines a new coordinate frame called $\underrightarrow{\mathcal{F}}_n$. It is important to note that this reference frame's x-axis will always point from segment i to $i+1$.

4. Compute the vector from the path point n to the vehicle frame, but expressed in the new coordinate frame, $\underrightarrow{\mathcal{F}}_n$ (see Figure 6.7(a)):

$$\mathbf{p}_n^{v,n} := \mathbf{C}_z(\alpha)\mathbf{p}_v^{v,n} = \mathbf{C}_z(\alpha)\left(-\mathbf{p}_v^{n,v}\right),$$

where $\mathbf{p}_v^{n,v}$ is simply constructed from the x- and y-components of $\mathbf{p}_v^{i,v}$. The lateral error is the y-component of this vector: $\varepsilon_L := \mathbf{p}_n^{v,n}(2)$.

5. The heading error is computed according to the difference between the vehicle's heading and the heading of the path segment coordinate frames, projected into the local 2D vehicle frame (see Figure 6.7(b)). A unit vector in the x direction from each path pose is projected into the vehicle frame to measure the errors:

$$\mathbf{e}_v^i := \mathbf{C}_{v0}\mathbf{C}_{0i}\begin{bmatrix}1\\0\\0\end{bmatrix}, \qquad \mathbf{e}_v^{i+1} := \mathbf{C}_{v0}\mathbf{C}_{0i+1}\begin{bmatrix}1\\0\\0\end{bmatrix}.$$

Then we compute the linear combination of the two errors according to

$$\varepsilon_H := (1-\lambda)\varepsilon_{H,i} + \lambda\varepsilon_{H,i+1},$$

where λ is the ratio of the vehicle position projected along the path segment and divided by the total path segment length: $\lambda := \frac{\mathbf{p}_n^{v,n}(1)}{\mathbf{p}_n^{n+1,n}(1)}$. The rational for using a linear combination of the path-node heading errors was to account for the possibility of large path segments, which are in fact used in a related project called *Network of Reusable Paths* (Stenning and Barfoot, 2011).

In addition to the above mentioned procedure for computing the heading and lateral errors, a look-ahead distance is also used to measure the heading error in front of the vehicle. This is done to allow the vehicle to gradually transition into a turn and to help reduce overshoot. As the vehicle is not limited to simply one linear speed, a set of target speeds were chosen and controller gains for each speed were determined through experimentation. This *gain scheduling*

Table 6.2: Path tracking control gains

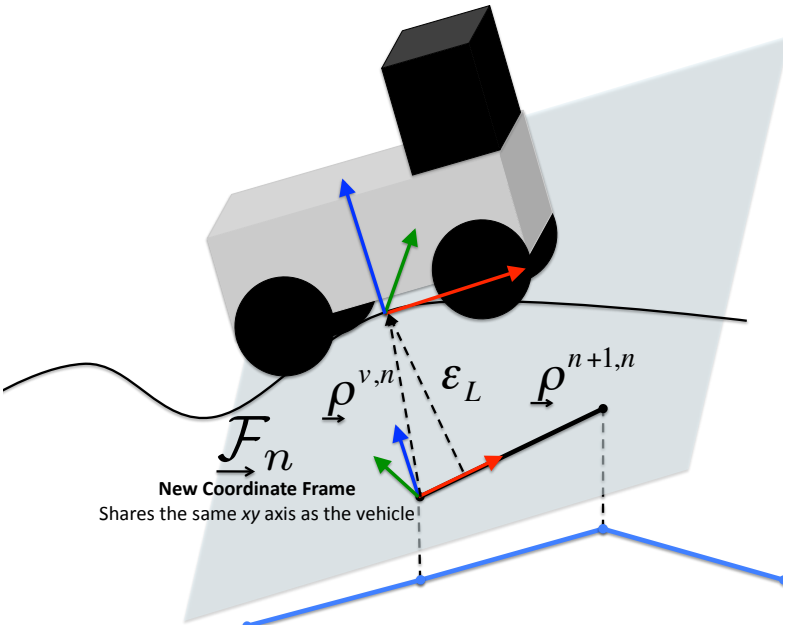
Speed [m/s]	Lateral Error Gain (ε_L)	Heading Error Gain (ε_H)	Look-Ahead Distance [m]
$\pm$ 0.35	0.28	2.50	0.50
$\pm$ 0.50	0.40	2.50	0.75

was an important aspect of improving the path tracker's performance characteristics, since its derivation uses a constant speed assumption. Table 6.2 provides the speed schedules and controller gains that were used for these experiments.

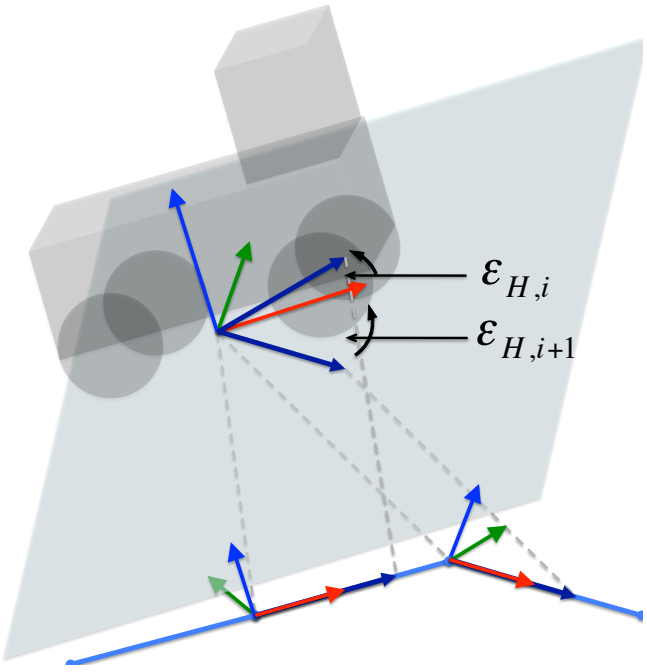
Determining what target speeds were appropriate for a given path was done by a *speed profiler*, which sweeps a window (3m in these experiments) along the reference path and computes the root-mean-squared incremental orientation changes within that window, according to the following

$$\phi := \sum_{i}^{i+N} \sqrt{\frac{\delta\boldsymbol{\theta}_i^T \delta\boldsymbol{\theta}_i}{N}},$$

where $\delta\boldsymbol{\theta}_i$ is the orientation change from pose i to $i-1$ expressed in Euler angles. For these experiments, if $\phi < 3.5^\circ$, then a speed of ± 0.50m/s is chosen, otherwise, the slower speed of ± 0.35m/s is chosen.



(a) The definition of a new coordinate frame to measure the lateral error.



(b) This figure shows how the heading error is measured.

Figure 6.7: Lateral and heading error calculations.

Chapter 7

Experiments

This chapter presents long-range VT&R field tests with a high-framerate Autonosys lidar. All tests were conducted at the Ethier Sand and Gravel pit in Sudbury, Ontario, Canada, as part of the Sudbury Lunar Analogue Missions — a joint venture between the Canadian Space Agency, the University of Western Ontario, MDA Space Missions, and the University of Toronto, to test a complete operations concept of a lunar sample and return mission. This site proved to be a very effective lunar analogue environment due to its lack of vegetation and sandy/rocky terrain (see Figure 7.1). In total, over 11km of autonomous driving was achieved and post-processed differential GPS (DGPS) was used for groundtruth. The experiment involved manually teaching a 1.1km route outdoors at approximately 7:45 pm in sunlight and autonomously repeating that route every 2-3 hours for 25 hours. What will follow is a description of the hardware used in this 25 hour experiment, followed by the experimental results.

7.1 Hardware Description

The mobile platform used in these experiments was a six wheeled, skid steered vehicle that has an articulated chassis with three individual pods, where the fore and aft pods can pitch and roll relative to the middle pod. The vehicle was equipped with a Thales DG-16 Differential GPS unit, an Autonosys LVC0702 lidar, and two Macbook Pro computers (one used to interface

Figure 7.1: Left: a GPS track of the 1154m taught route in the Etheir Sand and Gravel pit in Sudbury, which proved to be an effective analogue environment due to its lack of vegetation and 3D terrain. Right: an image of the ROC6 field robot during an autonomous repeat traverse.

with the Autonosys to port-forward data packets and the other for all of the lidar processing and control). In addition to the onboard DGPS, another DGPS was setup as a static base station to allow for real-time kinematic corrections. The Circular Error Probability (CEP) for these differential GPS units is 40cm[1]. An image of the field robot equipped with its sensors is shown in Figure 7.2.

The Autonosys LVC0702 is a high-framerate amplitude modulated continuous wave (AMCW) lidar that measures the shift in phase of a reflected sinusoidally-modulated laser signal in order to compute range. Using this phase shift, ϕ, as well as the modulation frequency, λ, one can compute the time of flight of the laser pulse, t, according to

$$t = \frac{\phi}{2\pi\lambda},$$

which can then be used to compute range. Intensity information is determined by the difference in amplitude between the emitted and returned signal. One of the benefits of AMCW lidar

[1]CEP is defined as the radius of a circle where 50% the data will fall.

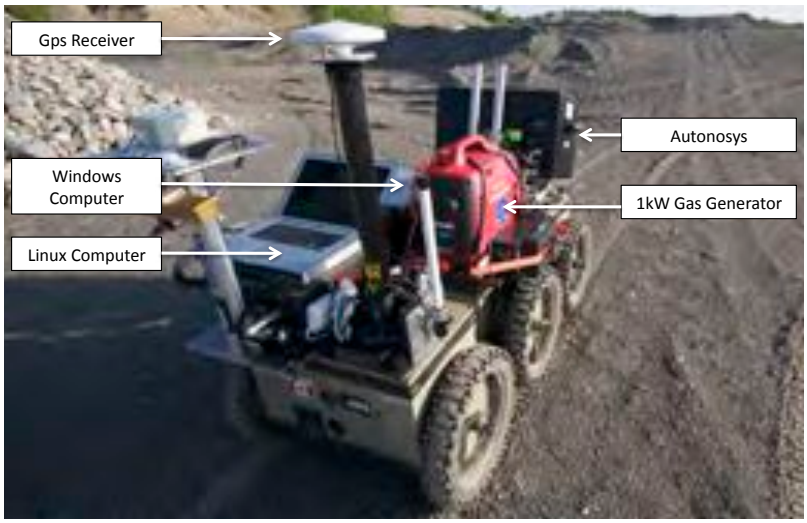


Figure 7.2: ROC6 field robot and its sensor configuration. The robot is equipped with the high-framerate Autonosys lidar at the front, a GPS receiver at the rear, a 1kW gas generator, and two laptop computers (the Windows computer is directly connected to the Autonosys and port-forwards raw data data to the Linux computer, which performs the localization in ROS).

versus pulsed TOF lidar is a larger dynamic range in intensity information, which results from how the different approaches detect the incoming laser signals. In general, most lidar detectors use an avalanche photodiode (APD), which applies a high reverse-bias voltage in order to increase the gain of the return signal (Francois, 2004). For good ranging accuracy, pulsed TOF lidars generally increase the bias voltage over time in order to amplify weak signals that have returned from long distances. However, when the bias voltage is increased too high, reflected light can saturate the detector, making the recovery of the original shape and size of the pulse difficult. Since the shape and size of the pulse is related to the reflectivity, or *intensity*, pulsed TOF sensors can lose some intensity information through this amplification process, but have the benefit of long-distance ranging. In contrast, AMCW lidar leave the bias voltage constant in order to maintain the sinusoidal shape of the modulated signal. However, the disadvantage of AMCW lidar versus pulsed TOF lidar is their smaller maximum range, which results from an ambiguity in phase after half of a wavelength (Wehr and Lohr, 1999).

The Autonosys achieves such high framerates (e.g., 10Hz) because of a patented mir-

ror assembly that combines a nodding and hexagonal mirror to scan a 45^oV/90^oH Field of View (FOV) with a pulse repetition rate (PRR) of 500,000 points/second. The basic premise behind the mirror design is to use a rotating hexagonal mirror for high-speed scanning in the horizontal direction and a nodding mirror to deflect the scan vertically where less angular speed is required. Nodding mirrors have the advantage of being able to collect all of the reflected light over their entire angular range, but the necessity to slow down and reverse direction at the edges of the FOV limits them to slow angular scan rates (ONeill et al. (2010)). In contrast, hexagonal mirrors rotating at constant rate allow for fast angular scanning with no reversal in mirror motion. However, rotating polygonal mirrors have the disadvantage that some light collection efficiency is lost near the edges of their FOV. Thus, by combining both types of mirrors, the LVC0702 is able to achieve a compromise between speed and collection efficiency. The LVC0702 provides 15-bit intensity information, has a maximum range of approximately 53.5m and can scan as fast as 10Hz; however, increasing the frame rate results in lower image resolutions.

For these experiments, the vertical field of view of the sensor was reduced from 45^o to 30^o at $\pm 15^o$ in order to capture 480x360 images at 2Hz. The rational behind restricting the vertical field of view was to increase the angular resolution in the vertical direction while trying to maintain a feasible scanning rate. Higher resolution in the vertical field of view is ideal since the spacing between elevation points projected onto a flat plane increases with distance. Thus, the goal was to reduce spatial aliasing as much as possible (this is discussed in more detail in section 8). In addition, the sensor was pitched downward 15^o in order to focus most of the scans towards the ground and not the sky.

7.2 Field Tests

A 1154m route was taught during sunlit conditions around 7:45 pm and autonomously repeated every 2-3 hours for a total of 10 runs, covering over 11km. It should be stressed that the lighting

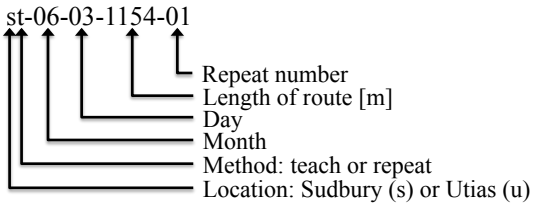


Figure 7.3: Teach and Repeat naming convention.

varied from full daylight to full darkness over the course of this experiment and the system was always matching to the full daylight conditions (i.e., the teach pass). The route was taught to resemble a realistic exploration mission, traversing to a number of dead-ends and thus requiring backtracking to explore new areas. Loop closures were also incorporated in order to make the dataset useful for future use, as loop closures are extremely important in SLAM.

As was done in Furgale and Barfoot (2010) and Royer et al. (2007), the performance of the system is evaluated using two different metrics: (i) the lateral error from the teach pass to the repeat pass measured with DGPS and (ii) the difference between the estimated lateral error to the teach pass and the actual lateral error. The first measure provides a metric for how well the entire closed-loop system is able to track the teach pass, while the second metric assesses the accuracy of the localization engine in estimating the lateral offsets to the teach pass. Two tables of results have been compiled. Table 7.1 provides the start time, end time, and distance covered autonomously for each of the 10 runs. Table 7.2 provides performance metrics, such as Root Mean Square (RMS) path error, max path error, and average VO/map match counts. The naming convention used is shown in Figure 7.3 and follows a similar convention as used in Furgale and Barfoot (2010).

Monitoring different off-nominal situations was an important aspect of these experiments. In particular, four different off-nominal modes were recorded and have been indicated on all figures: (i) VO failures, (ii) map localization failures, (iii) map and VO failures, and (iv) manual interventions due to the vehicle being stuck or failing to localize against the map within a reasonable time limit (e.g., 3 minutes). It should be noted that failure modes (i)-(iii) are fully recoverable and in almost all of these cases, the system was able to continue autonomously

Table 7.1: Autonomy rates.

Tag	Start Time (hh:mm:ss)	End Time (hh:mm:ss)	Distance Covered Autonomously (%)
sr-06-03-1154-01	23:03:27	00:03:39	100
sr-06-04-1154-02	01:26:53	02:50:34	99.85
sr-06-04-1154-04	05:00:28	05:56:26	99.91
sr-06-04-1154-05	09:47:12	10:57:13	99.48
sr-06-04-1154-06	11:51:36	13:20:19	98.49
sr-06-04-1154-07	14:15:54	15:35:51	99.46
sr-06-04-1154-08	16:25:05	17:32:41	100
sr-06-04-1154-09	18:24:19	19:18:41	100
sr-06-04-1154-10	20:31:06	21:37:36	100
sr-06-04-1154-11	22:58:43	23:50:06	100

Table 7.2: Performance results. RMS lateral error is the measured lateral offset using DGPS. RMS localization error is the difference between the estimated and the measured lateral error.

Tag	RMS Lateral Error [cm]	Max Lateral Error [cm]	RMS Loc. Error [cm]	Max Loc. Error [cm]	Avg. No. VO Matches per frame	Avg. No. Map Matches per frame
sr-06-03-1154-01	8.2	34.6	7.8	45.8	218	219
sr-06-04-1154-02	7.9	30.0	8.0	-39.5	163	101
sr-06-04-1154-04	12.7	42.2	12.0	56.5	166	92
sr-06-04-1154-05	17.7	-88.8	17.5	108.2	119	66
sr-06-04-1154-06	23.9	-84.8	23.9	87.2	115	71
sr-06-04-1154-07	15.6	-109.2	15.5	-122.2	160	82
sr-06-04-1154-08	11.1	52.4	11.6	-95.4	163	80
sr-06-04-1154-09	15.9	-59.0	15.5	-93.5	185	84
sr-06-04-1154-10	15.3	-53.5	14.6	-64.2	182	80
sr-06-04-1154-11	12.5	51.7	11.9	-116.1	170	83

using the strategies outlined in the previous section. Figures 7.4-7.13 show the groundtruth trajectory of the vehicle for every repeat pass as well as the match counts and error plots. It should be noted that repeat pass 3 has been omitted because of a failure to complete due to a software issue during the run (i.e., this failure had nothing to do with the VT&R algorithm, but required a complete system reboot).

Additionally, it is also important to note that there are some limitations to the accuracy of the groundtruth used to asses the performance of the system. Firstly, there is a discrepancy in the measured difference between the repeat pass runs and the teach pass run due to the fact that different satellites are observed in each run because of the long periods of time between the various trials. Secondly, our DGPS has a CEP of 40cm, which is actually quite large compared to the level of accuracy of the VT&R technique; this is especially true since the rover was always moving so there was no averaging of points in a stop-and-go fashion. Thirdly, due to physical constraints on the platform, the GPS receiver was mounted on the opposite end of the robot from the actual sensor, meaning that the estimated lateral error and the measured lateral error could be different depending on the orientation of rover pods[2]. These factors should be kept in mind when examining the difference between DGPS measured lateral error and the estimated lateral error.

What will follow is a brief summary of each repeat pass run, focusing on the observed performance of the system, how it recovered from off-nominal modes, and the environmental changes that took place. It is important to stress that in the following figures, only those off-nominal modes that are labeled/coloured *manual control* actually required human assistance.

1. This was the most successful run in terms of the least number of VO/map failures, the highest number of VO and map matches, and a low RMS tracking error of 8cm. It should also be noted that this traverse was done in complete darkness, demonstrating the benefit of lidar as the teach pass was done during sunlight. This run was completed fully autonomously without any manual interventions.

[2]Onboard inclinometer data was available, but is extremely noisy and was not used in processing these results.

2. This was another successful run, achieving a low RMS tracking error for most of the traverse. However, midway through, it began to rain, which proved to be a challenging environmental change that would affect map matching for the rest of the experiments. Nearing the end of the traverse, failures to localize against the map resulted in manual interventions to move the vehicle along the path until it relocalized. In total, only 1.67m was traversed manually, making this run 99.86% autonomous. It should be noted that due to GPS dropouts, groundtruth for approximately 200 meters of the traverse was lost, so error is only reported on the remaining portion. In addition, the last section of GPS appears to have a systematic offset that was not observed during the actual experiment. Unfortunately, this type of GPS failure was observed in a couple of our datasets. Appealing to the honour system, we have decided to discount these sections from computing the error, as such large deviations were not observed during the actual experiment.

3. This traverse was not completed due to a software bug that caused the system to crash, as well as a power shutdown on our groundstation that resulted in a loss of base station GPS data. As a result, this traverse was cancelled, but it should be stressed that this was in no way related to the algorithm itself.

4. Although the scene appearance had changed due to the rain and map match counts were low compared to the first repeat pass, this was another successful repeat pass that began in pitch black conditions and ended at sunrise. As was the case with run 2, we had encountered GPS dropouts and lost groundtruth for the first 200m of the traverse. At the second direction switch indicated on Figure 7.6, the vehicle's rear wheels had sunk into the soft soil and required a manual intervention to move it out of this stuck position; this was a mechanical issue, not an algorithm issue. Aside from this one failure, the algorithm worked fully autonomously and completed the run with low error.

5. Prior to this repeat run, it had rained quite heavily, which undoubtedly changed the reflectivity of the soil. As a result, during this repeat run, the system was unable to match

against the map effectively, losing localization in a number of areas and requiring manual control to carry it past some areas where it could not localize. Having said this, the system was still able to repeat this route almost fully autonomously ($\sim 99.5\%$). During one section of the run where map localization was lost, VO actually carried the system passed this troublesome area despite a large lateral error and guided the vehicle back on the path allowing the system to relocalize (see Figure 7.7). It should be noted that GPS dropouts similar to runs 2 and 4 were encountered, meaning that error was only reported on roughly 900m of the traverse.

6. The sun was extremely strong during this repeat pass and interestingly, this run had the largest number of VO and map failures, and required manual interventions at two direction switches (see Figure 7.8). Although lidar is supposed to be lighting invariant, McManus et al. (2011) demonstrated that matching lidar intensity images roughly 12 hours apart yielded the least frame-to-frame matches, which is not entirely surprising, since lidar sensors must filter any external light. Having said this, 98.5% of the route was traversed fully autonomously; thus, even in the worst case, this system is still extremely robust. It should also be noted that the groundtruth lateral error measurements appear to have a noticeable offset that was not observed during the actual experiment. This is either due to spurious GPS points at the beginning of the run that have caused an alignment issue, or due to the fact that different satellites actually measured the position of the vehicle from the teach pass to the repeat pass, which could have introduced a bias in the estimates.

7. Repeat pass 7 had similar error profiles as number 6, with localization failures encountered at the direction switches indicated in Figure 7.9. In total 99.47% of the route was repeated fully autonomously.

8. Despite a GPS dropout during the early portion of the traverse, this repeat pass was accomplished fully autonomously, with an RMS path error of 11cm.

9. This was a completely successful autonomous run without any manual interventions.

10. This run was fully autonomous except for the second dead end (see Figure 7.10), where a software bug resulted in a failure to update the nearest branch keyframe. Referring to Section 6, this VT&R architecture works by updating the pose estimate via frame-to-frame VO and then localizing against the nearest keyframe. In this case, the nearest keyframe was not being updated and as a result, the map matching continued to fail. Eventually the system recovered and relocalized at the apex of the direction switch, which is where the problem originally surfaced. Since this failure mode was the result of a software bug and does not represent a shortcoming of the VT&R algorithm, we have not included this manual intervention in our autonomy rate calculation.

11. This run was done in the dark and completed fully autonomously. Unfortunately for this run, our GPS estimate was not very accurate and there is an offset near the beginning of the run that led to the final position being over a meter away from the true position when aligned at the start (this section has been omitted from the plots). Instead, the end position of the GPS track was aligned with the start position as they were in fact coincident within 10cm, which appears to be more representative of the true run (evidenced by the fact that the vehicle repeated the route fully autonomously). Admittedly, this could introduce a potential bias in the results, but this is an unfortunate situation where the groundtruth was in fact less accurate than the method it is benchmarking.

As the results indicate, this lidar-based VT&R system is extremely effective, achieving centimeter-level accuracy and driving in its own tracks most of the time. In total, over 11km was repeated over 25 hours of operation with an autonomy rate of 99.7% by distance, demonstrating the robustness and effectiveness of appearance-based lidar as a substitute for passive, camera-based systems. The next chapter will provide a summary and discussion of these results, focusing on why these various failure modes occurred and what improvements can be made to prevent them in the future.

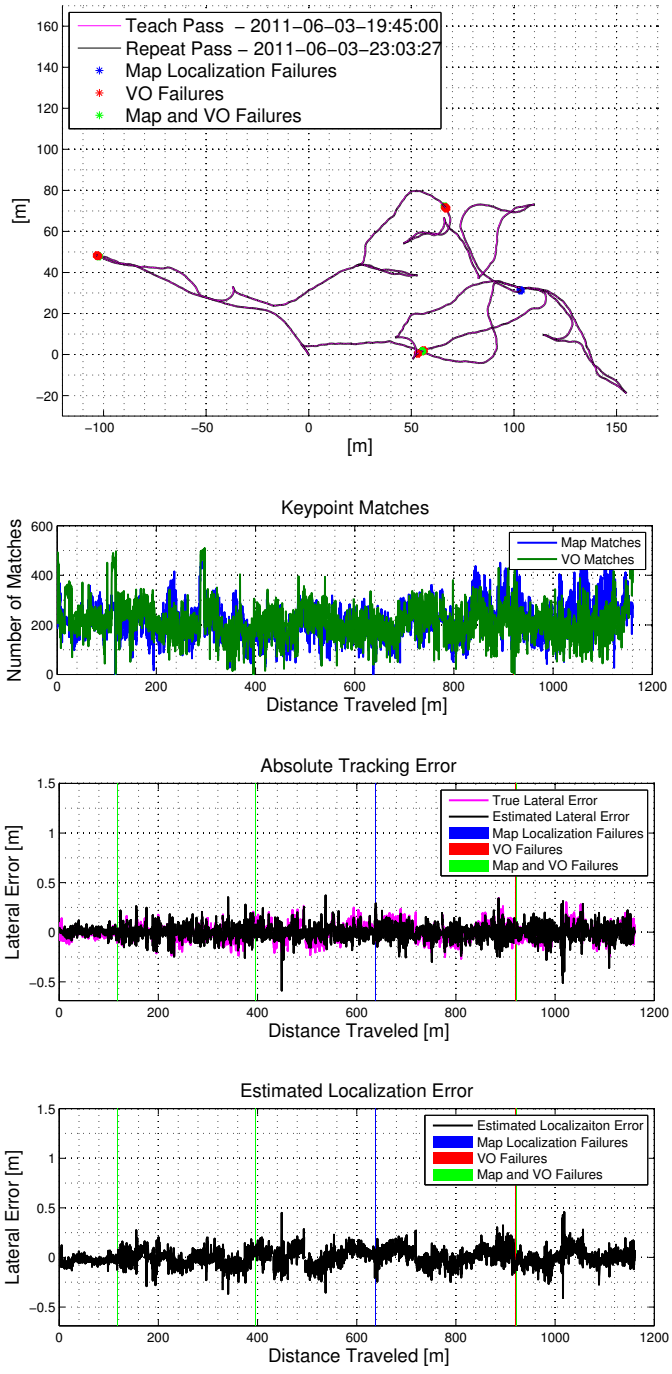


Figure 7.4: Repeat pass 1 results.

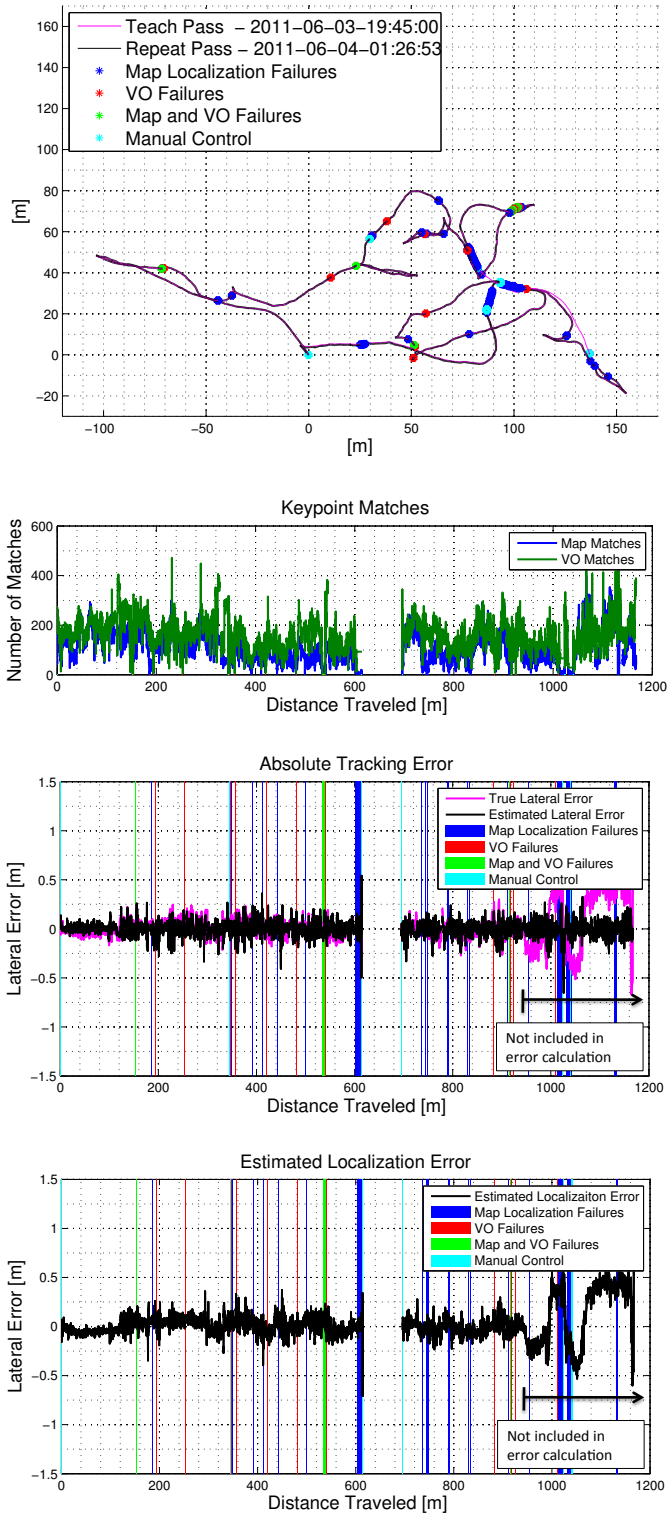


Figure 7.5: Repeat pass 2 results.

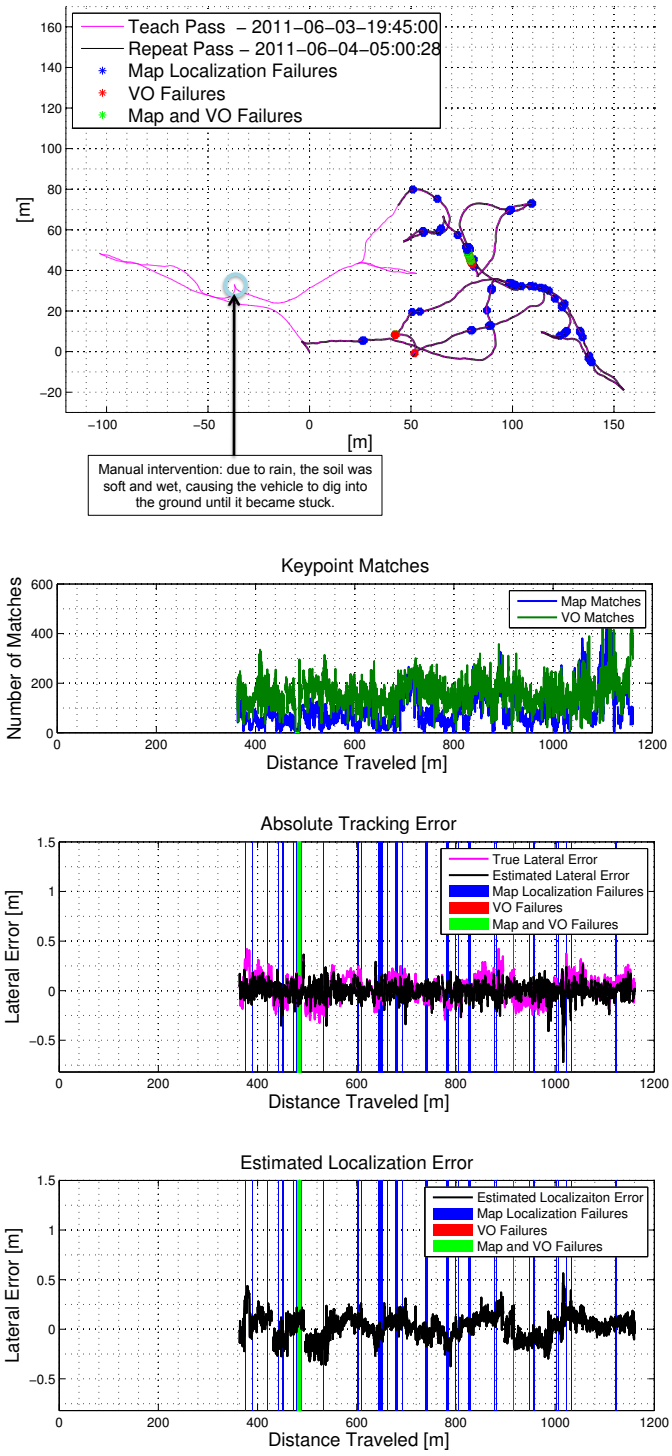


Figure 7.6: Repeat pass 4 results.

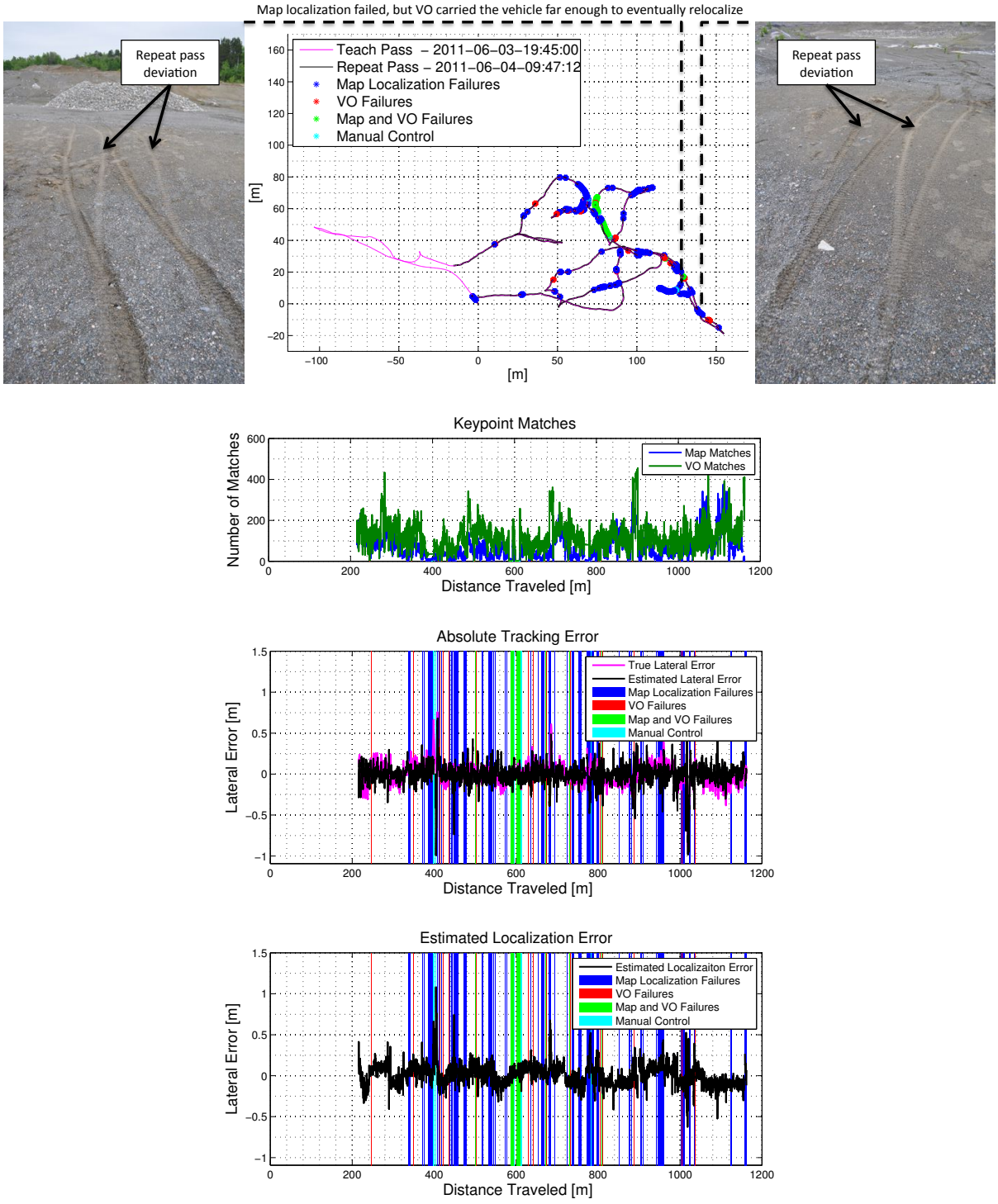


Figure 7.7: Repeat pass 5 results.

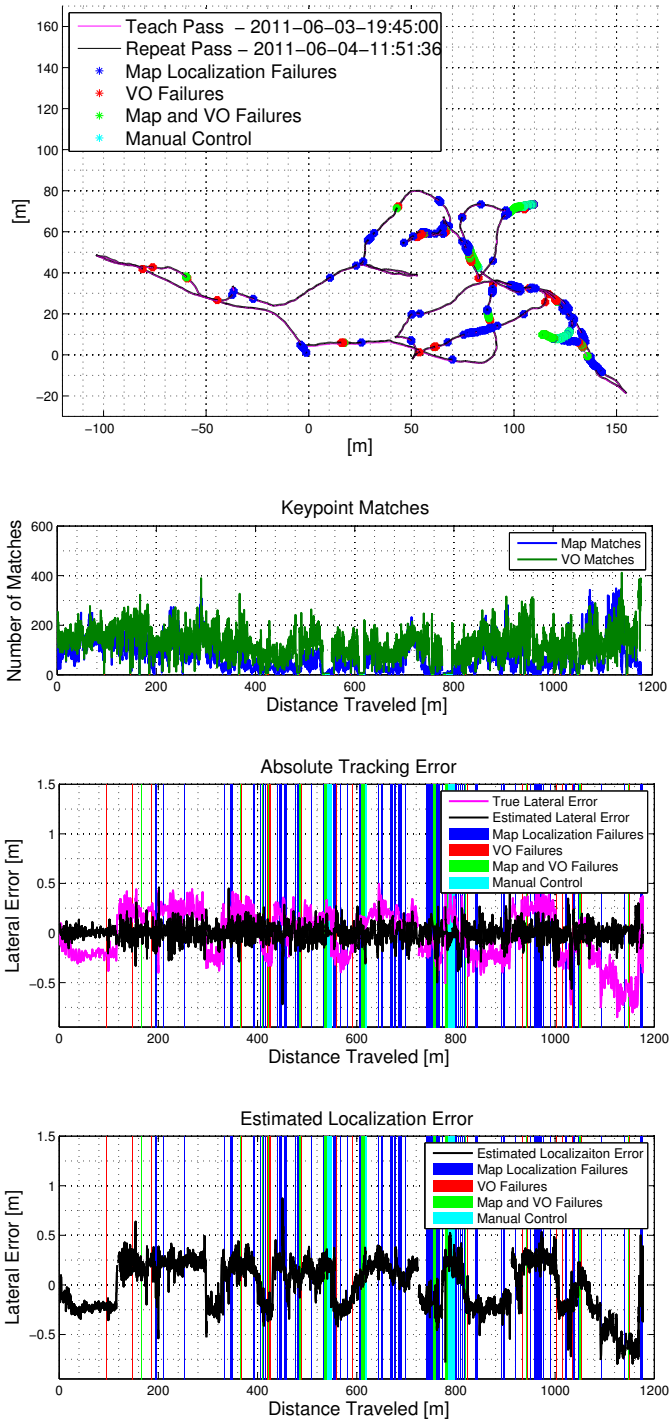


Figure 7.8: Repeat pass 6 results.

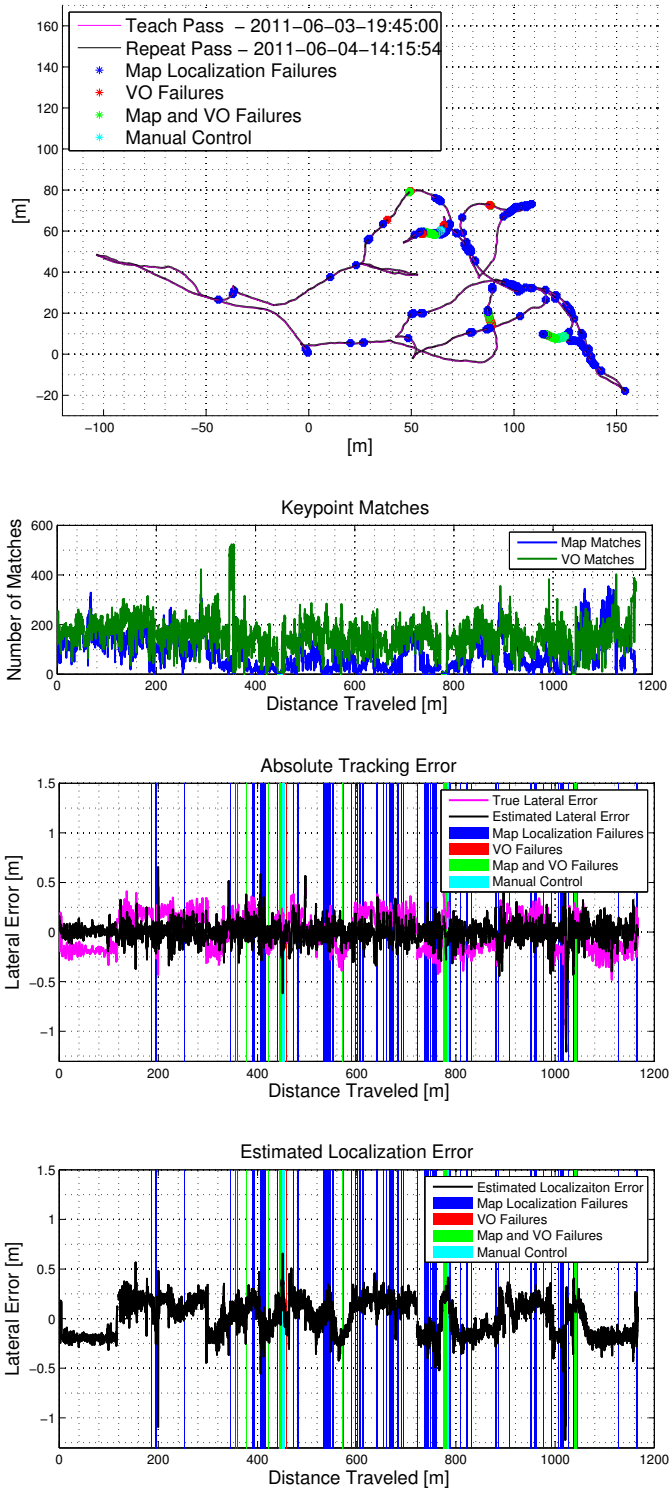


Figure 7.9: Repeat pass 7 results.

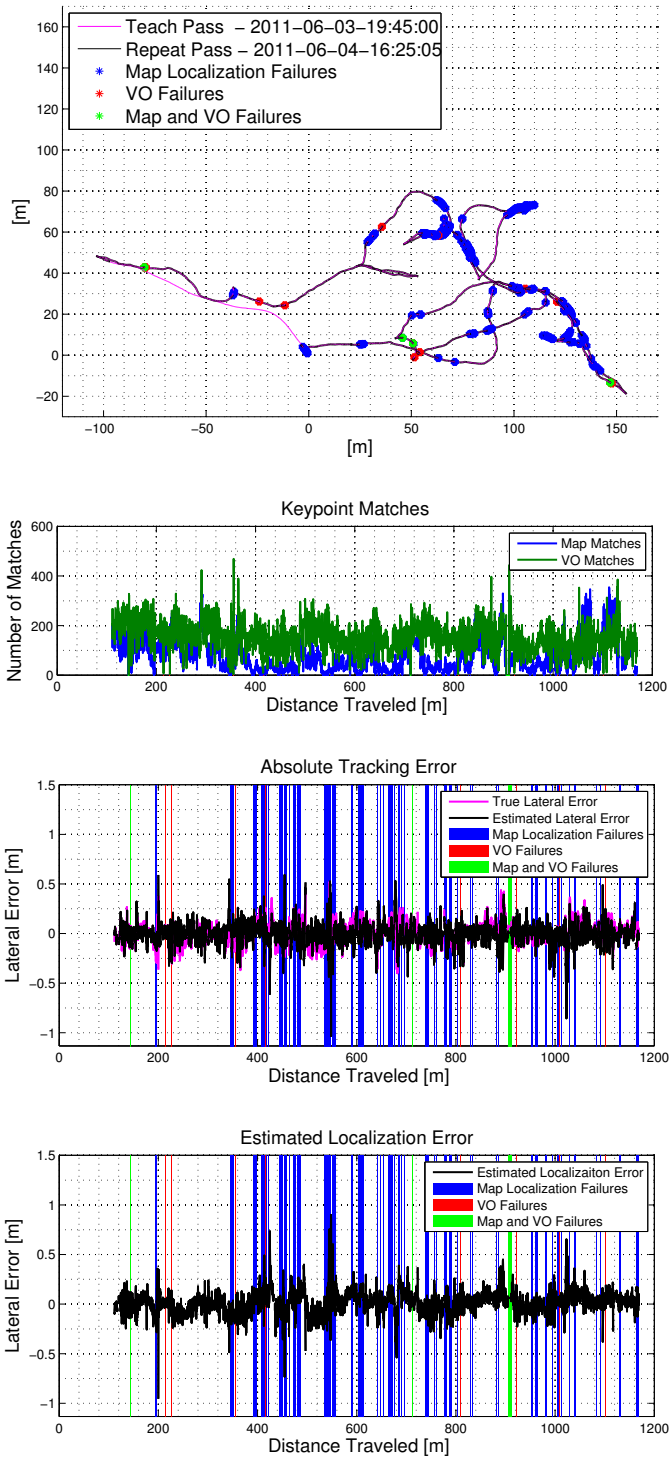


Figure 7.10: Repeat pass 8 results.

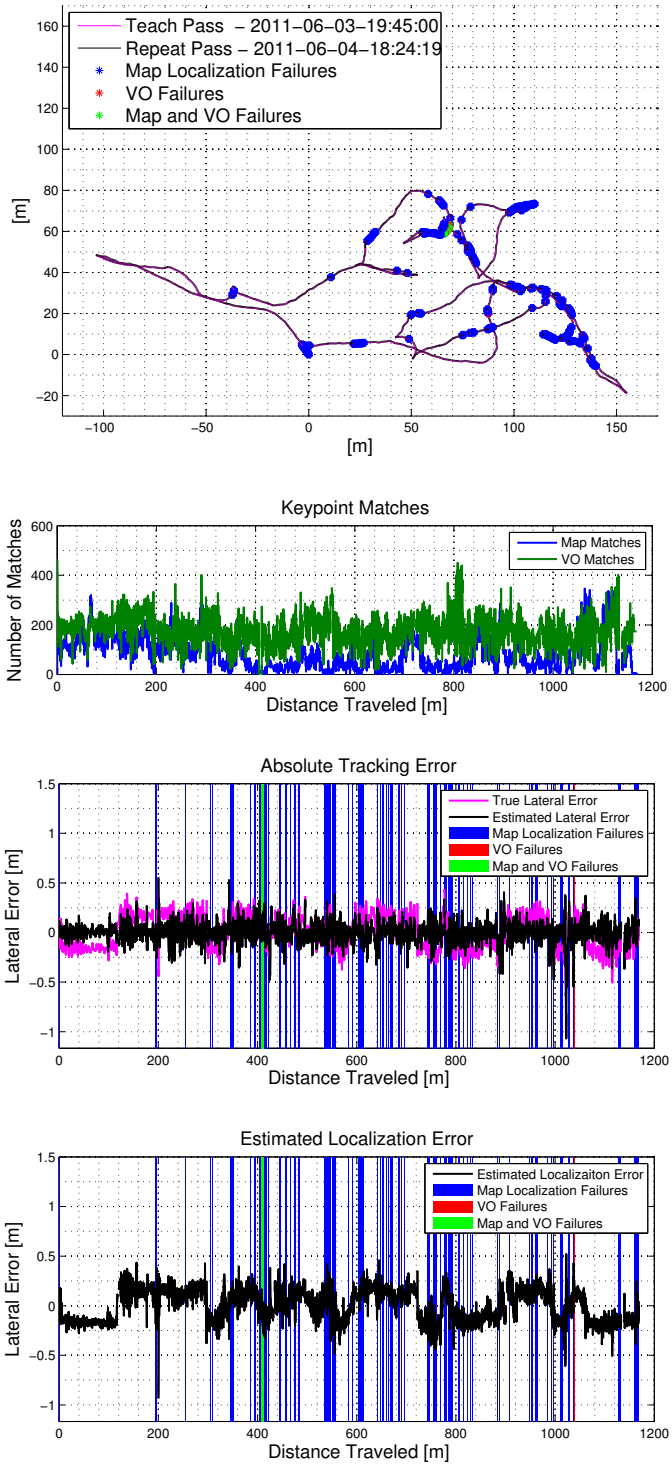


Figure 7.11: Repeat pass 9 results.

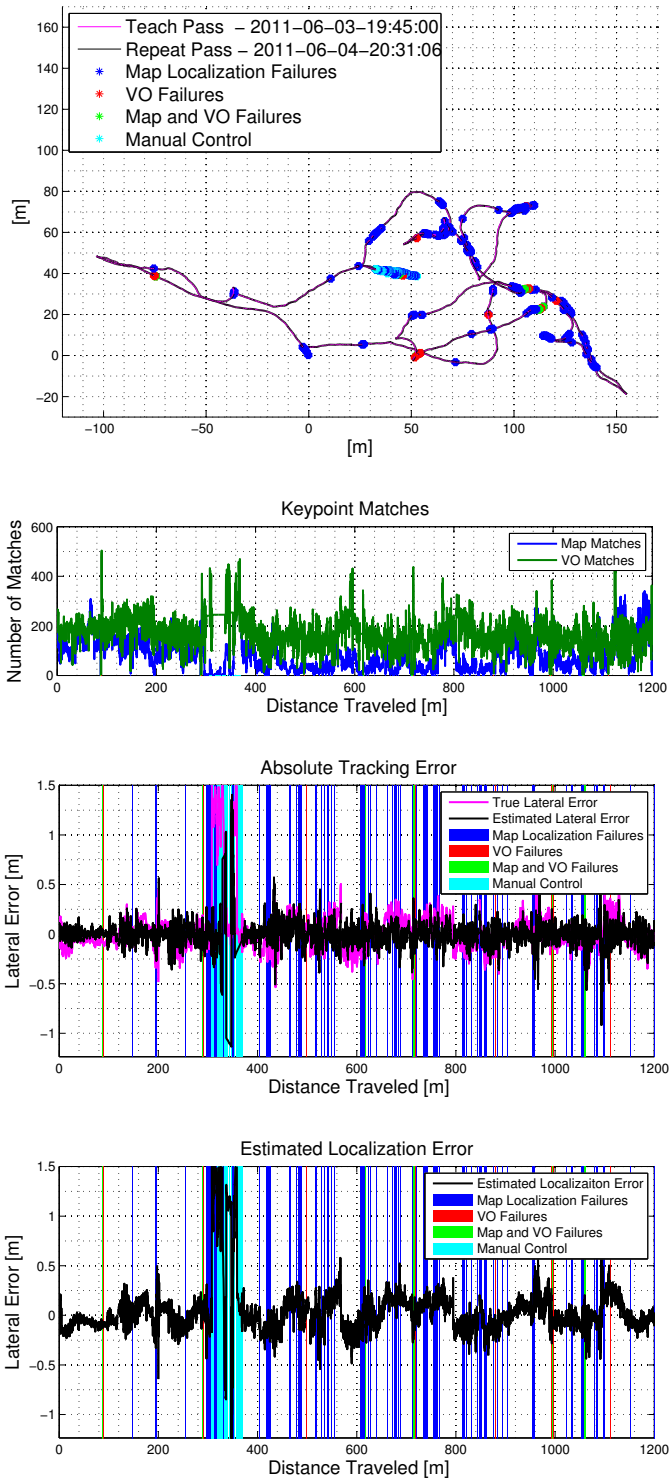


Figure 7.12: Repeat pass 10 results.

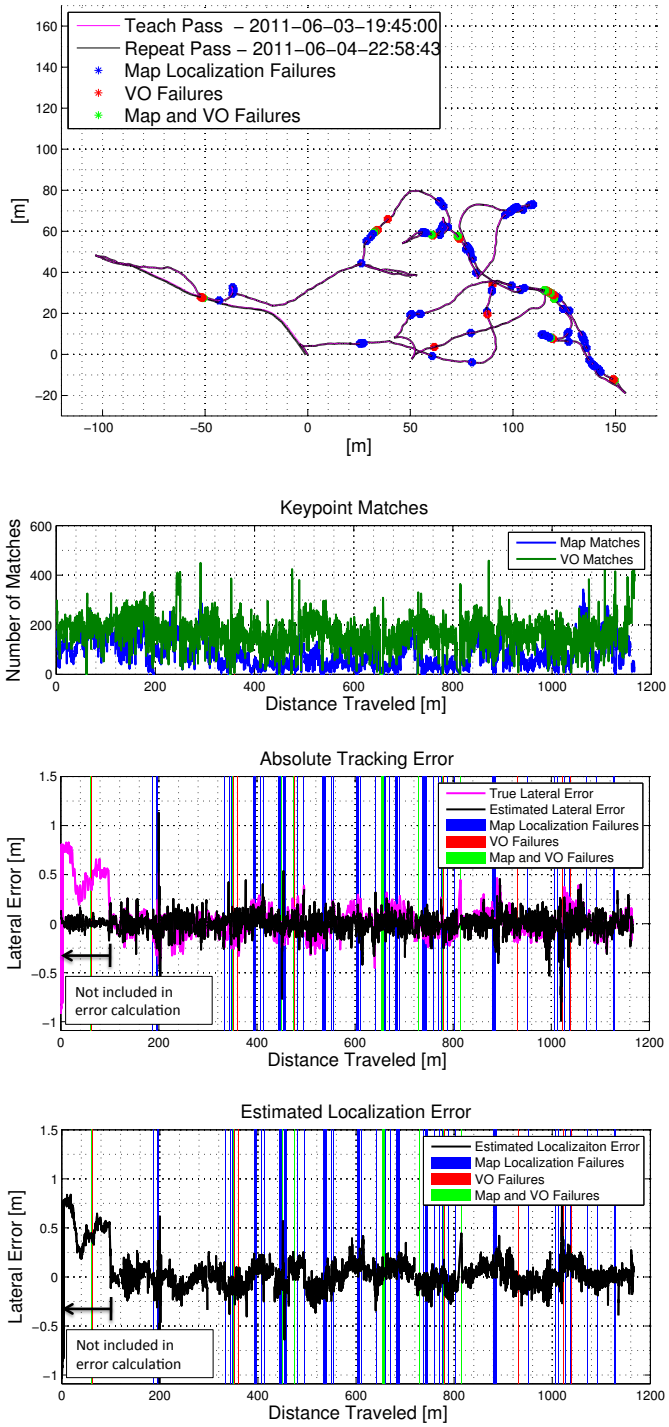


Figure 7.13: Repeat pass 11 results.

Chapter 8

Discussion

To summarize the results, the following plots were compiled. Figure 8.1 shows the GPS track with all of the failure modes from all the runs superimposed, Figure 8.2 shows the average number of VO/map matches for each run, Figure 8.3 shows the total number of different failure modes, and Figure 8.4 shows the average errors for all of the runs.

Beginning with Figure 8.1, it is clear that localization failures occurred over a large portion of the map and with a high frequency in a number of texture-poor areas. Examples of what these failures look like in image space are provided in Figure 8.5. VO failures generally occur from motion blur (i.e., large motions between frames which can cause feature matching to fail). This is also true when both VO and map matching fail simultaneously; however, another major cause of VO/map matching failures is significant data loss, which is an implementation issue and not an algorithm issue. Referring to the map/VO and VO failures on Figure 8.5, it is clear that the search window size for finding candidate matches may have been too small, which resulted in low match counts for images with a large level of distortion. Failure to match against the map generally occurs either due to significant viewpoint changes from path tracking deviations or because of significant changes in scene appearance (the latter case can be seen in Figure 8.5, as the scene lacked sufficient texture for a successful match).

Referring to the VO/map matches, it is interesting to note that repeat run 1 had the highest number of VO/map matches, followed by a small dip near noon and then approaching a rel-

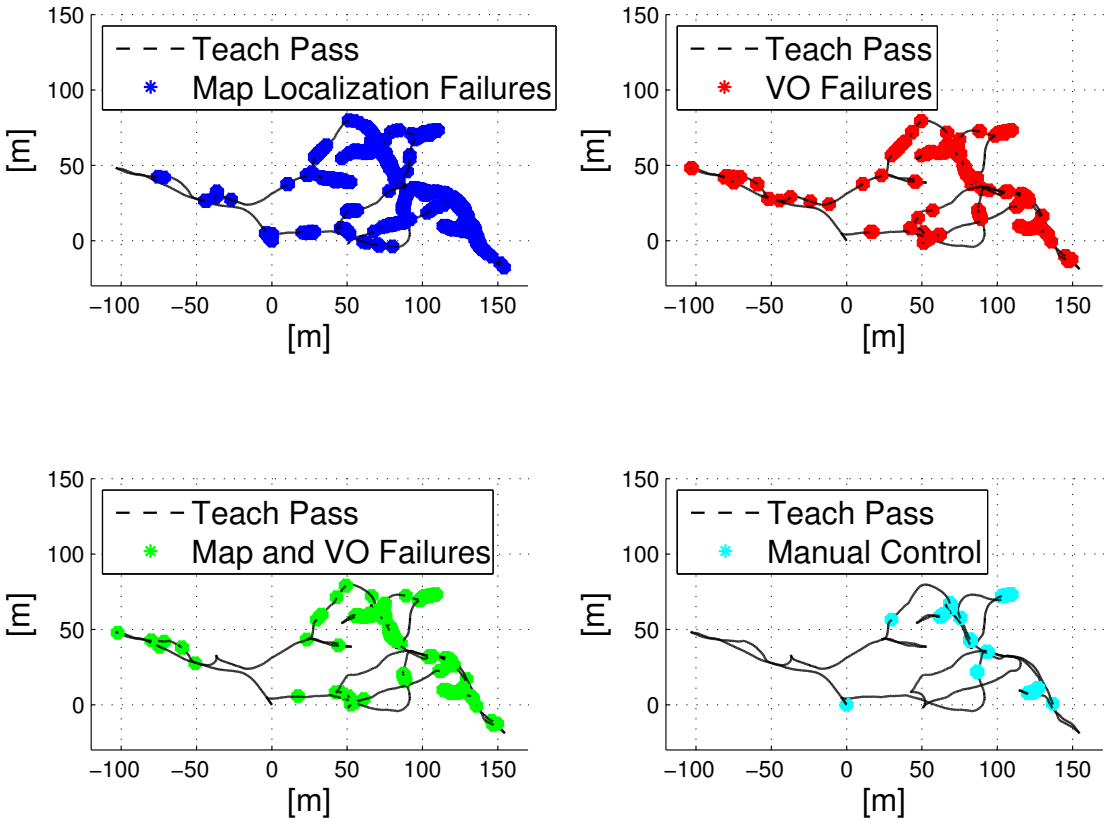


Figure 8.1: Plot of all failure modes superimposed on GPS track. Note that only the manual control failures were not recovered automatically.

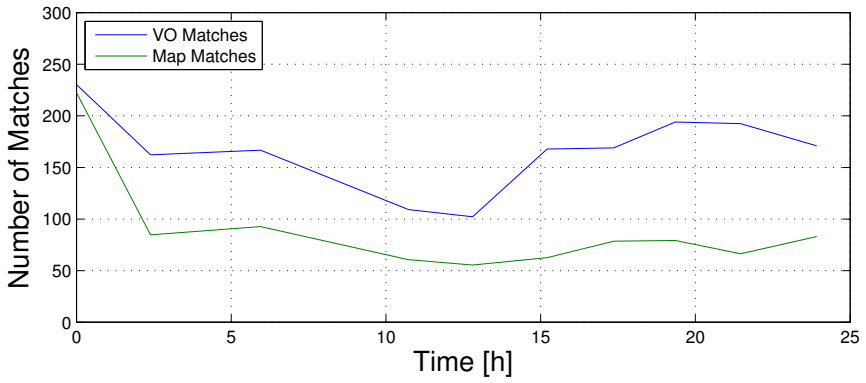


Figure 8.2: Average number of VO matches and map matches per repeat run.

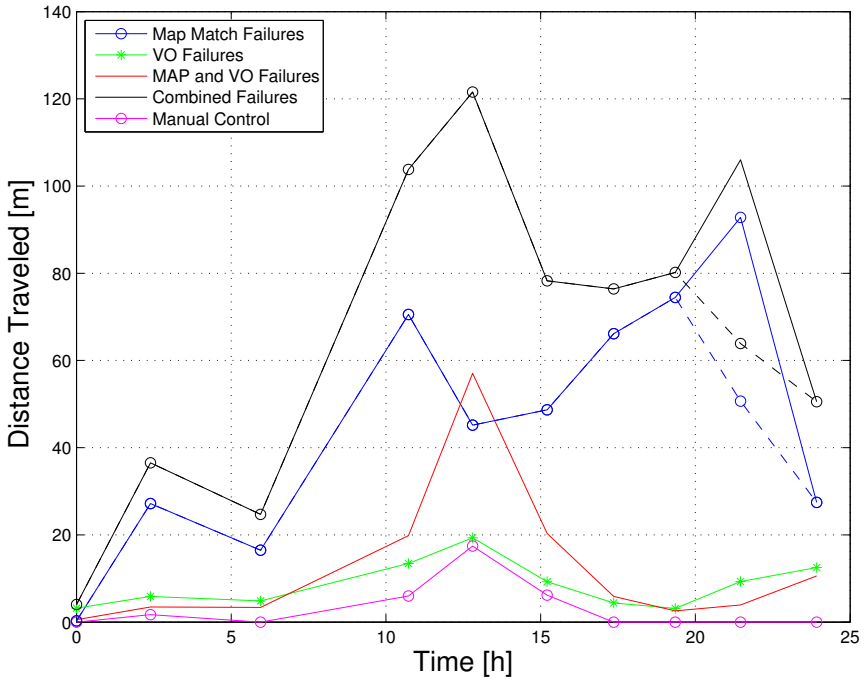


Figure 8.3: Total number of failures measured by distance traveled versus the time since the teach pass. A dashed line has also been drawn to indicate the number of map match failures for run 10, discounting the software bug that caused localization failures in the second dead-end. The black line is simply the sum of all the failures, indicating that matching images roughly 12 hours apart yields the worst repeating performance.

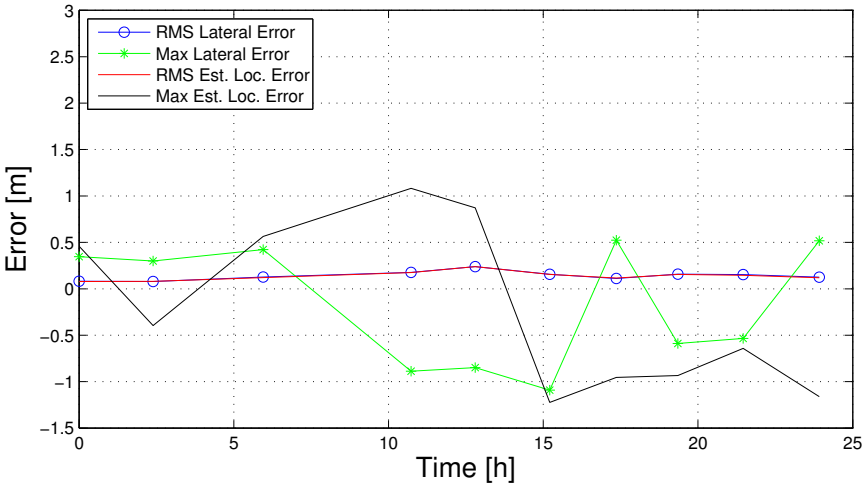


Figure 8.4: Average error metrics for each repeat run.

atively constant value for the rest of the runs. It was expected that map matches would drop over time for the reasons explained above; however, it is difficult to explain why VO matches dropped, since VO is based on matching current and previous frames. One possible explanation could be due to the heavy rain fall, which changed the reflectivity of the soil and resulted in less texture overall (towards hour 19, it was noticed that the ground had dried significantly) . The dip during noon could also be explained by the fact that the sun was very direct and strong during this time, which could affect the lidar somewhat, as it must filter incoming light.

Referring to the number of failures in Figure 8.3, it seems clear that repeating the route nearly 12 hours apart results in the highest number of VO and VO/map failures and the largest number of failures overall. Note that the large map failure peak at the 23-hour mark occurred during repeat run 10 and is due to the software issue that caused map failures in the second dead-end. Discounting that event, and noting the drop-off in the number of matches around the 12-hour mark (see Figure 8.2), the results confirm what was observed by McManus et al. (2011), which is that the lowest number of matches occur between lidar intensity images separated by 12 hours (Figure 8.3 shows the map failure profile disregarding the software bug). Again, this is not a surprising result given the fact that lidar sensors will be somewhat affected by ambient lighting conditions; however, even in the worst case (run 6), the system was able to repeat the taught route with an autonomy rate of 99.5%, which is a feat that could not be accomplished with a passive sensor under the dramatic lighting changes studied here.

In many of the cases where human intervention was required, it was because the system failed to localize against the map and continued running on VO until it reached the localization failure threshold of 3m. As discussed in Section 6, when the system fails to localize against the map, it will use VO until it reaches a distance threshold of 3m, after which, the system will stop the vehicle and begin its search for the map. This recovery method worked well with the stereo-based system by Furgale and Barfoot (2010) since their VO was reasonably accurate up to 50m, allowing the system to traverse past feature-poor areas. However, in the case of this system, metric VO is very inaccurate, meaning that the system cannot trust VO over longer

distances. This is due to a powerful assumption that was silently used in Section 3.

The assumption that was used is that when forming the image stack, all of the pixels in the image arrive at the same instant in time, which is of course false, since the laser is continually scanning while moving. This means that range values from one part of the image arrive later than others, creating a distorted view of the scene that results in a biased estimate (see Figure 8.6 for an illustration of the distortion in one of the intensity images). In theory, since the timestamp of each laser reading is known, a time correction could be applied to compensate for this distortion, but this will be the focus of future work. As demonstrated, long-range accurate VO is not necessary for VT&R, which is one of the major strengths of the technique. However, the lack of accurate VO means that the system is less robust to map localization failures since VO will not be able to accurately guide the vehicle forward in hopes of relocalizing against the map. This was indeed the case for the regions where manual control was needed, as the inaccurate VO was unable to successfully bring the vehicle beyond these feature-poor regions. Only in a couple of cases was VO successful in carrying the system beyond 3m without localizaing against the map (see Figure 7.7). Clearly, applying motion compensation for better VO would be a significant improvement to the system's robustness.

Regarding the error plots, it is clear that the groundtruth measurements were not reliable, since the difference in estimated lateral error and measured lateral error are of the same order of magnitude. Again, this is the result of a number of factors mentioned earlier, such as comparing GPS runs at different times of the day (meaning that different satellites measured each run) and measuring the lateral error at a different location than the estimated lateral error. Regardless, the measured lateral error from each repeat pass to the teach pass was still on the order of centimeters and given the very high autonomy rate for each run, it is clear that the localization engine worked well enough to validate the effectiveness of this technique over the largest changes of ambient lighting. As qualitative evidence, Figure 8.7 shows an image sequence of one section of an autonomous retro-traverse, which shows the vehicle driving in its own tracks.

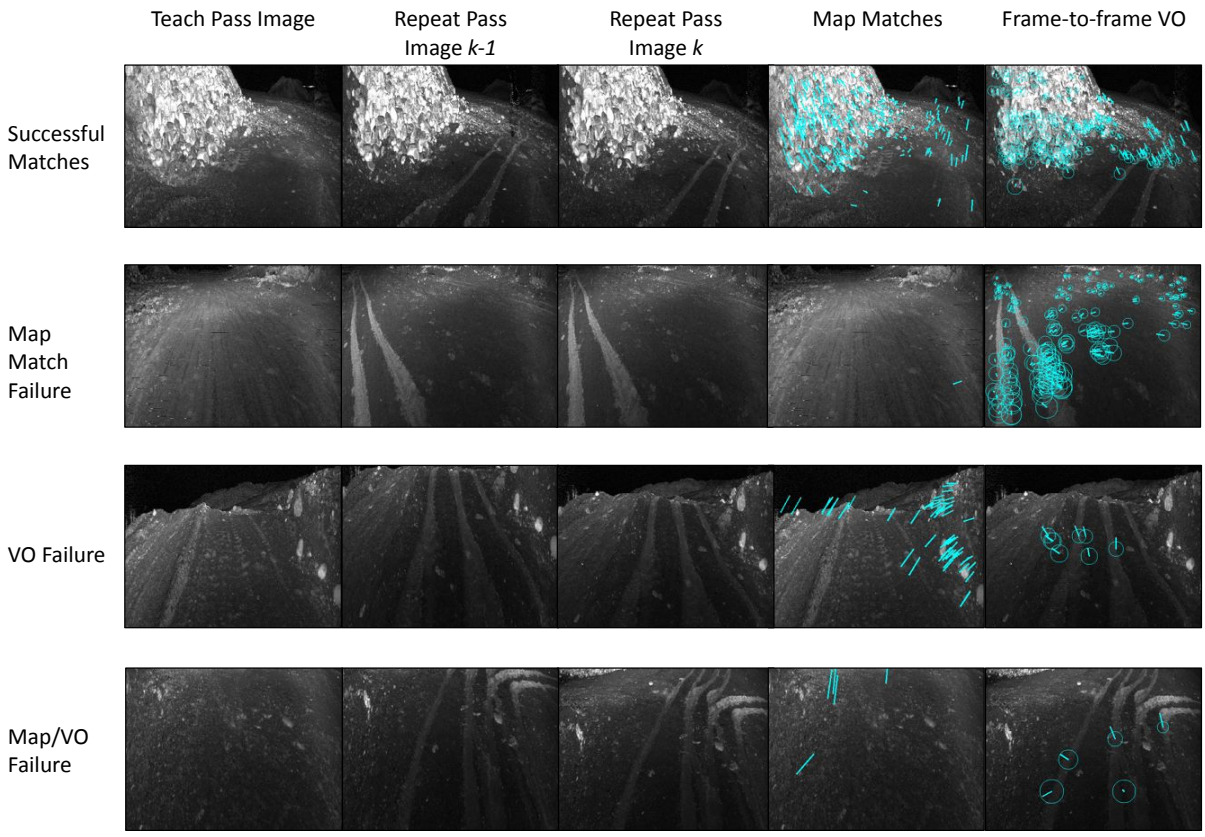
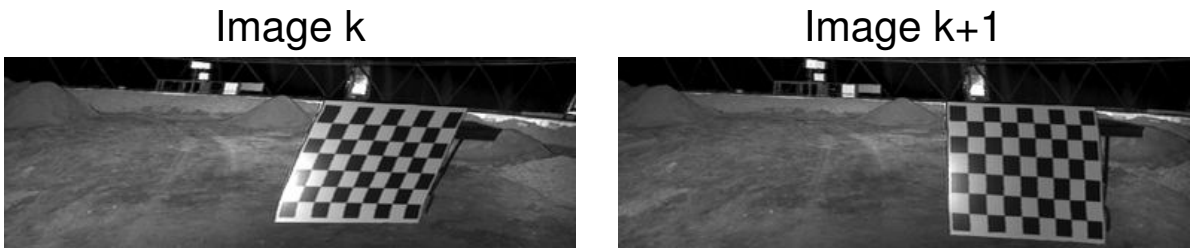
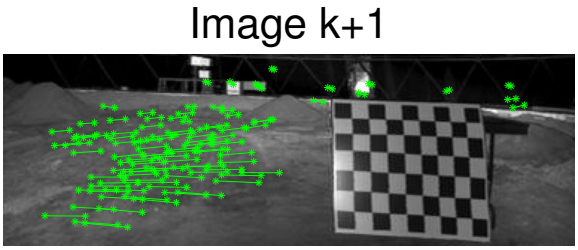


Figure 8.5: Examples of various failure modes. Column 1: nearest teach pass image. Column 2: the previous repeat pass image, denoted as image $k-1$. Column 3: the current repeat pass image, denoted as image k. Column 4: matching the current repeat pass image to the teach pass image (i.e., matching column 1 to column 3). Column 5: frame-to-frame VO matches for the repeat pass (i.e., matching column 2 to column 3). Images have not been scaled according to the $90^\circ \times 30^\circ$ FOV due to size constraints. Images are 480×360 and were captured at 2Hz while in motion. Circles in the VO feature tracks are proportional to the landmarks range (i.e., closer landmarks have larger circles).



(a) In this image, we can see some warping in the checkerboard, which was due to rotational motion during scanning.

(b) In the next frame, the checkerboard appears upright without any significant distortion.



(c) Keypoint tracks on image $k + 1$. Although we did not track any keypoints on the checkerboard, we were able to track more than 100 keypoints on the ground.

Figure 8.6: These images show some of the distortion resulting from scanning and moving at the same time. In this case, the vehicle was moving at approximately 0.5m/s and the Autonosys was capturing at 2Hz, meaning that each image was collected over approximately 0.25m of travel. Interestingly, for matching keypoints in image space, this distortion did not turn out to be a bottleneck in the system. However, the distortion in the range image affects the metric accuracy of VO, which does result in a biased motion estimate, since the geometry of the scene is being warped. For accurate VO, motion compensation must be applied.

Figure 8.7: Images of an autonomous repeat, where the robot can be seen driving in its own tracks off in the distance. The bottom row shows some of the tracks that were repeatedly traversed over all 10 runs.

Chapter 9

Conclusion

This thesis has detailed the design, implementation, and testing of a lighting-invariant Visual Teach and Repeat (VT&R) system that combines fast and effective appearance-based computer vision techniques with a state-of-the-art high-framerate lidar sensor. The main purpose of this research was to design a method that would enable long-range autonomous retro-traverses for planetary sample and return missions; however, there are many other applications for this technique that extend beyond the space domain (e.g., patrolling, underground mining, and convoying). By using lidar as the primary sensor, this system is able to avoid one of the more challenging aspects of visual perception in outdoor environments: dynamic lighting conditions, which proved to be a limiting factor for Furgale and Barfoot (2010). Through long-range field tests in a planetary analogue environment, the system's robustness and overall effectiveness was demonstrated on over 11km of travel, 99.7% of which was traversed fully autonomously.

This work in novel in a number of ways. Firstly, it is the only VT&R system that uses an appearance-based lidar approach for motion estimation. Secondly, it is the first VT&R system to introduce the concept of an *augmented keyframe* as the local map, which allows for a simplistic system architecture that can build maps and localize online. Lastly, the system was fully tested in a planetary analogue environment over multiple kilometers in rough 3D terrain. This is a worthy contribution in and of itself, as it highlights the many strengths and

weaknesses of the technique in a realistic setting. One of the major lessons learned from these field tests is that accurate, metric VO would be extremely beneficial to carry the robot past feature-poor zones without deviating significantly from the path. This will require some form of motion compensation to handle the fact that scanning and moving at the same time produces a distorted intensity and range image, which in turn, produces a drift in the estimate. Developing a motion compensation method for such a data-dense sensor is certainly a problem worth exploring.

Looking forward, it is clear that lighting invariance is only one piece of the puzzle, as a number of challenges still remain. In particular, dealing with path obstructions and finding a way to function in environments that gradually change over time (e.g., erosion, changing seasons) are two open problems that should be addressed. The former issue deals with local path planning and repair, while the latter issue is a much more interesting problem, as it deals with the concept of long-term autonomy, which is quickly becoming a popular topic in robotics.

Nonetheless, this thesis work has demonstrated a proof-of-concept lighting-invariant lidar-based VT&R system that runs online and was able to autonomously traverse over multiple kilometers in challenging 3D terrain. Hopefully, it will just be a matter of time before the space-rated technologies catch up to the modern-day terrestrial laser scanners and computing, as this technique will lend itself very well to sample and return missions on Mars or exploration missions to permanently-shadowed lunar craters.

Chapter 10

Acronyms

VO Visual Odometry

VT&R Visual Teach and Repeat

lidar Light Detection and Ranging

SLAM Simultaneous Localization and Mapping

ICP Iterative Closest Point

TOF Time of Flight

GPU Graphics Processing Unit

KLT Kanade-Lucas-Tomasi

RANSAC Random Sample and Consensus

CEP Circular Error Probability

RMS Root Mean Square

GPS Global Positioning System

FOV Field of View

ROS Robot Operating System

SURF Speeded-Up Robust Features

SIFT Scale Invariant Feature Transform

Bibliography

Abymar, T., Hartl, F., Hirzinger, G., Burschka, D., and Frohlich, C. (2007). Automatic registration of panoramic 2.5d scans and color images. In *Proceedings of the International Calibration and Orientation Workshop EuroCOW*, number 54, Castelldefels, Spain.

Allen, P., Feiner, S., Troccoli, A., Benko, H., Ishak, E., and Smith, B. (2004). Seeing into the past: Creating a 3d modeling pipeline for archaeological visualization. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*.

Antoine Maintz, J. and Vierger, M. (1997). A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36.

Argyros, A., Bekris, K., and Orphanoudakis, S. (2001). Robot homing based on corner tracking in a sequence of panoramic images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 3–10.

Argyros, A., Bekris, K., Orphanoudakis, S., and Kavraki, L. (2005). Robot homing by exploiting panoramic vision. *Auton. Robots*, 19(1).

Bae, K.-H. and Lichti, D. (2008). A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63:36–54.

Bajracharya, M., Maimone, M., and Helmick, D. (2008). Autonomy for mars rovers: Past, present, and future. *IEEE Computer Society*.

Baumgartner, E. and Skaar, S. (1994). An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–502.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359.

Bekris, K., Argyros, A., and Kavraki, L. (2006). Exploiting panoramic vision for bearing-only robot homing. *Imaging Beyond the Pinhole Camera*.

Besl, P. J. and McKay, N. D. (1992). A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2).

Biesiadecki, J. J., Baumgartner, E. T., Bonitz, R., Cooper, B., Hartman, F., Leger, P., Maimone, M., Maxwell, S., Trebi-Ollennu, A., Tunstel, E. W., and Wright, J. R. (2006). Mars exploration rover surface operations - driving opportunity at meridiani planum. *IEEE Robotics & Automation Magazine*, 13(2).

Blanc, G., Mezouar, Y., and Martinet, P. (2005). Indoor navigation of a wheeled mobile robot along visual routes. *IEEE International Conference onRobotics and Automation*.

Bohm, J. and Becker, S. (2007). Automatic marker-free registration of terrestrial laser scans using reflectance features. In *Proceedings of the 8th Conference on Optical 3D Measurement Techniques*, pages 338–344, Zurich, Switzerland.

Booij, O., Terwijn, B., Zivkovic, Z., and Krose, B. (2007). Navigation using an appearance based topological map. *IEEE International Conference on Robotics and Automation*.

Borrmann, D., Elseberg, J., Lingemann, K., Nuchter, A., and Hertzberg, J. (2008). Globally consistent 3d mapping with scan matching. *Robotics and Autonomous Systems*, 56:130–142.

Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139.

Bosse, M. and Zlot, R. (2009). Continuous 3d scan-matching with a spinning 2d laser. *IEEE International Conference on Robotics and Automation*, pages 4312–4319.

Brock, J., Wright, C., Sallenger, A., Krabill, W., and Swift, R. (2002). Basis and methods of nasa airborne topographic mapper lidar surveys for coastal studies. *Journal of Coastal Research*, 18(1):1–13.

Brown, D. C. (1958). A solution to the general problem of multiple station analytical stereotriangulation. Rca-mtp data reduction technical report no. 43, Patrick Airforce Base, Florida.

Cartwright, B. and Collett, T. (1983). Landmark learning in bees: experiments and models. *Journal of Comparative Physiology*, 151:521–543.

Cartwright, B. and Collett, T. (1987). Landmark maps for honeybees. *Biological Cybernetics*, 57(1/2):85–93.

Chen, Z. and Birchfield, S. (2006). Qualitative vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, United States.

Courbon, J., Blanc, G., Mezouar, Y., and Martinet, P. (2007). Navigation of a non-holonomic mobile robot with a memory of omnidirectional images. *Workshop: Planning, Perception and Navigation for Intelligent Vehicles*.

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.

Dempster, A. (1967). Upper and lower probabilitis induced by a multivalued mapping. *The Annals of Statistics*, 28:325–339.

Diosi, A., Remazeilles, A., Segvic, S., and Chaumette, F. (2007). Outdoor visual path following experiments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4265–4270.

Dold, C. and Brenner, C. (2006). Registration of terrestrial laser scanning data using planar patches and image data. In *Proceedings of the ISPRS Commission V Symposium 'Image Engineering and Vision Metrology'*, volume XXXVI, Dresden, Germany.

Donoghue, D., Watt, P., Cox, N., and Wilson, J. (2007). Remote sensing of species mixtures in conifer plantations using lidar height and intensity data. *Remote Sensing of Environment*, 110:509–522.

Droeschel, D., Holz, D., Stuckler, J., and Behnke, S. (2010). Using time-of-flight cameras with active gaze control for 3d collision avoidance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, United States.

Durrant-Whyte, F. and Bailey, T. (2006). Simultaneous localization and mapping: part i. In *Robotics & Automation Magazine*, volume 13, pages 99–110. IEEE.

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

Francois, B. (2004). Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1):231–240.

Franz, M., Sch´olkopf, B., and B´ulthoff, H. (1998). Where did i take that snapshot? scene-based homing by image matching. *Biological Cybernetics*, 79.

Fraundorfer, F., Engels, C., and Nister, D. (2007). Topological mapping, localization and navigation using image collections. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, pages 3872–3877.

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

Geman, S. and McClure, D. (1987). Statistical method for tomographic image reconstruction. In *Proceedings of the 46th Session of the International Statistical Institute, Bulletin of the ISI*, volume 52, pages 5–21.

Goedeme, T., Nuttin, M., Tuytelaars, T., and Van-Gool, L. (2007). Omnidirectional vision based topological navigation. *International Journal of Computer Vision*, 74(3):219–236.

Goedeme, T., Tuytelaars, T., Van Gool, L., Vanacker, G., and Nuttin, M. (2005). Feature based omnidirectional sparse visual path following. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Guivant, J., Nebot, E., and Baiker, S. (2000). Autonomous navigation and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583.

Haala, N., Reulke, R., Thies, M., and Aschoff, T. (2004). Combination of terrestrial laser scanning with high resolution panoramic images for investigations in forest applications and tree species recognition. In *Proceedings of the ISPRS working group*, volume V/1, IAPRS - XXXIV (Part 5/W16).

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Holfe, B. and Pfeifer, N. (2007). Correction of laser scanning intensity data: data and model-driven approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(6):415–433.

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

Husmann, K. and Pedersen, L. (2008). Strobe lit high dynamic range stereo imagery for dark navigation. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Hollywood, United States.

Jones, S., Andresen, C., and Crowley, J. (1997). Appearance based processes for visual navigation. In *Proceedings of the IEEE Int. Conference on Intelligent Robots and Systems*.

Kidono, K., Miura, J., and Shirai, Y. (2002). Autonomous visual navigation of a mobile robot using a human-guided experience. *Robotics and Autonomous Systems*, 40(2-3):124–132.

Koch, O. and Teller, S. (2009). Body-relative navigation using uncalibrated cameras. In *Proceedings of the International Conference on Robotics and Automation*, Kyoto, Japan.

Koch, O., Walter, M., Huang, A., and Teller, S. (2010). Ground robot navigation using uncalibrated cameras. In *Proceedings of the International Conference on Robotics and Automation*, Anchorage, Alaska, United States.

Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., and Fua, P. (2010). View-based maps. *The International Journal of Robotics Research*, 29(8):941–957.

Kretschmer, U., Abymar, T., Thies, M., and Frohlich, C. (2004). Traffice construction analysis by use of terrestrial laser scanning. In *Proceedings of the ISPRS working group VIII/2 "Laser-Scanners for Forest and Landscape Assessment"*, volume XXXVI, pages 232–236, Freiburg, Germany.

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

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, volume 2, San Francisco, California, United States.

Luzum, B., Starek, M., and Slatton, K. (2004). Normalizing alsm intensities gem center report - rep 2004-07-001. Technical report, University of Florida, United States.

Mails, E., Chaumette, F., and Boudet, S. (1999). 2-1/2-d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2).

Marshall, J., Barfoot, T., and Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6-7):400–421.

Matsumoto, Y., Ikeda, K., and Inaba, M. Inoue, H. (1999). Visual navigation using omni-directional view sequence. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 1, pages 317–322, Kyongju, South Korea.

Matsumoto, Y., Inaba, M., and Inoue, H. (1996). Visual navigation using view-sequenced route representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 83–88.

Matsumoto, Y., Sakai, K., Inaba, M., and Inoue, H. (2000). View-based approach to robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1702–1708, Takamatsu, Japan.

May, S., Fuchs, S., Malis, E., Nuchter, A., and Hertzberg, J. (2009). Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11-12):934–965.

McManus, C. (2009). Lidar-based teach and repeat for planetary exploration. *Technical Report (TR-2009-CM002). Institute for Aerospace Studies, University of Toronto*.

McManus, C., Furgale, P., and Barfoot, T. (2011). Towards appearance-based methods for lidar sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China.

Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., Haehnel, D., Hilden, T., Hoffmann, G., Huhnke, B., Johnston, D., Klumpp, S., Langer, D., Levandowski, A., Levinson, J., Marcil, J., Orenstein, D., Paefgen, J., Penny, I., Petrovskaya, A., Pflueger, M., Stanek, G., Stavens, D., Vogt, A., and Thrun, S. (2008). Junior: The stanford entry in the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part 2*, 25(9):569–597.

Neira, J., Tardos, J., Horn, J., and Schmidt, G. (1999). Fusing range and intensity images for mobile robot localization. *IEEE Transactions on Robotics and Automation*, 15(1):76–84.

Nistér, D. (2003). An efficient solution to the five-point relative pose problem. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 195–202.

Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, New York City, New York, United States.

Ohno, T., Ohya, A., and Yuta, S. (1996). Autonomous navigation for mobile robots referring pre-recordedimage sequence. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 672–679.

ONeill, J., Moore., W., Williams, K., and Bruce, R. I. (2010). Scanning system for lidar. United States patent US 0053715 A1.

Rabbani, T. and van den Heuvel, F. (2005). Automatic point cloud registration using constrained search for corresponding objects. In *Proceedings of 7th Conference on Optical 3-D Measurement Techniques*, pages 177–186.

Rekleitis, I., Bedwani, J.-L., and Dupuis, E. (2007). Over-the-horizon, autonomous navigation for planetary exploration. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2248–2255.

Richardson, A. and Rodgers, M. (2001). Vision-based semi-autonomous outdoor robot system to reduce soldier workload. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) 4364*, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, pages 12–18, Orlando, Florida, United States.

Royer, E., Lhuillier, M., Dhome, M., and Lavest, J. (2007). Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 74(3).

Schenker, P., Huntsberger, T., Pirjanian, P., Baumgartner, E., and Tunstel, E. (2003). Planetary rover developments supporting mars exploration, sample return and future human-robotic colonization. *Autonomous Robots*, 14:103–126.

Se, S., Ng, H.-K., Jasiobedzki, P., and Moyung, T.-J. (2004). Vision based modeling and localization for planetary exploration rovers. In *Proceedings of the 55th International Astronautical Congress*, Vancouver, Canada.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Sibley, G., Mei, C., Reid, I., and Newman, P. (2010). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The Int. Journal of Robotics Research*, 29(8):958–980.

Simhon, S. and Dudek, G. (1998). A global topological map formed by local metric maps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1708–1714, Victoria, BC, Canada.

Stenning, B. and Barfoot, T. (2011). Path planning on a network of paths. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montanta, United States.

Strasdat, H., Montiel, J., and Davison, A. (2010). Real-time monocular slam: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, Anchorage, Alaska, United States.

Surmann, H., Nuchter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198.

Tang, L. and Yuta, S. (2001). Vision based navigation for mobile robots in indoor environment

by teaching and playing-back scheme. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 3072– 3077, Seoul, Korea.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.

Thrun, S., Fox, D., and Burgard, W. (2000). A real-time algorithm for mobile robot mapping with application to multi robot and 3d mapping. In *Proceedings of the IEEE Int. Conference on Robotics and Automation*, pages 321–326, San Francisco, California, United States.

Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Kelly, A., Likhachev, M., McNaugton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S. Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., and Ziglar, J. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part 1*, 25(8):425–466.

Vardy, A. and Oppacher, F. (2003). Low-level visual homing. In *Proceedings of the 7th European Conference on Artificial Life*, pages 875–884.

Šegvić, S., Remazeilles, A., and Diosi, A. Chaumette, F. (2009). A mapping and localization framework for scalable appearance-based navigation. *Computer Vision and Image Understanding*, 113(2):172–187.

Wehr, A. and Lohr, U. (1999). Airborne laser scanning - an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54:68–82.

Weingarten, J. W., Gruner, G., and Siegwart, R. (2004). A state-of-the-art 3d sensor for robot

navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*, volume 3, pages 2155–2160, Lasuanne, Switzerland.

Wettergreen, D., Jonak, D., Kohanbash, D., Moreland, S., Spiker, S., and Teza, J. (2009). Field experiments in mobility and navigation with a lunar rover prototype. In *Proceedings of the 7th Int. Conf. on Field and Service Robotics*, Cambridge, Massachusetts, United States.

Wulf, O., Nuchter, A., Hertzberg, J., and Wagner, B. (2008). Benchmarking ubran six-degree-of-freedom simultaneous localization and mapping. *Journal of Field Robotics*, 25(3):148–163.

Ye, C. and Bruch, M. (2010). A visual odometry method based on the swissranger sr4000. In *Proceedings of the SPIE - Unmanned Systems Technology XII*, volume 7692.

Yoshitaka, H., Hirohiko, K., Akihisa, O., and Shin'ichi, Y. (2006a). Map building for mobile robots using a sokuiki sensor-robust scan matching using laser reflection intensity-. In *Proceedings of the SICE-ICASE Int. Joint Conf.*, pages 5951–5956, Bexco, Busan, Korea.

Yoshitaka, H., Hirohiko, K., Akihisa, O., and Shin'ichi, Y. (2006b). Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor. In *Proceedings of the 32nd Annual Conf. on IEEE Industrial Electronics*, Paris, France.

Yuan, F., Swadzba, A., Philippsen, R., Engin, O., Hanheide, M., and Wachsmuth, S. (2009). Laser-based navigation enhanced with 3d time-of-flight data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Kobe, Japan.

Zhang, A. M. and Kleeman, L. (2009). Robust appearance based visual route following for navigation in large-scale outdoor environments. *The International Journal of Robotics Research*, 28(3).

Zitova, B. and Flusser, J. (2003). Image registration methods: A survey. *Image and Vision Computing*, 21:977–1000.