MAP–CENTRIC VISUAL DATA ASSOCIATION ACROSS SEASONS IN A NATURAL ENVIRONMENT

A Dissertation
Presented to
The Academic Faculty

by

Shane Griffith

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
December 2019

MAP–CENTRIC VISUAL DATA ASSOCIATION ACROSS SEASONS IN A NATURAL ENVIRONMENT

Approved by:

Professor Cédric Pradalier, Adviser
School of Interactive Computing
Georgia Tech–Lorraine

Professor Frank Dellaert, Co-Adviser
School of Interactive Computing
Georgia Institute of Technology

Professor Anthony Yezzi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Tucker Balch
School of Interactive Computing
Georgia Institute of Technology

Professor Charles Isbell
School of Interactive Computing
Georgia Institute of Technology

Date Approved: October 30, 2019

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support, the help, and the input of many organizations and people. Financial support for the work in this dissertation was provided by the Lorraine Region, France, GeorgiaTech-Lorraine, and the Georgia Institute of Technology. I would also like to thank the administrators of the School of Interactive Computing and GeorgiaTech-Lorraine for providing invaluable tips and resources towards a more seamless boundary between the Atlanta and the Metz campuses. I would also like to acknowledge the time my committee spent critically reviewing this work. All of these shared efforts were essential to the success of this dissertation.

The beginnings of this work started with my advisor Prof. Cédric Pradalier. I arrived in France after the unmanned boat already captured several visual surveys. I undertook the work from the goal of building and using a spatiotemporal representation of the robot's observations. Cedric provided this structure, much of the framework for moving forward, and many helpful ideas along the way. A large part of the contributions in this dissertation can be attributed to him.

This dissertation also includes efforts of my colleagues Georges Chahine and Paul Drews, and my co-advisor Prof. Frank Dellaert. They provided help collecting surveys, creating manuscripts, and identifying relevant literature. In my meetings with Prof. Dellaert, he always had concise pieces of advice that brought a lot of value to the work.

This dissertation is also in large part due to the research foundations provided by several others. I have had many other academic advisors who provided significant one-on-one time, including Alex Stoytchev, Andrea Thomaz, Charles Isbell, Daji Qiao,

and Nicola Elia. I also learned from my supervisors Shila Gulati, Stefan Holzer, and Alex Trevor while on internships at Bosch and Fyusion.

I also have several people to thank for good conversations and nice company through the program, including: Marc Carroll, Rick Swette, Tucker Hermans, Brian Goldfain, Ankur Agarwal, Sarah Selim, Jon Scholz, Michael Gielniak, and Arya Irani.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Vision is one of the primary sensory modalities of animals and robots, yet among robots it still has limited power in natural environments. Dynamic processes of Nature continuously change how an environment looks, which work against appearance–based methods for visual data association. As a robot is deployed again and again, the possibility of finding correspondences diminishes between surveys increasingly separated in time. This is a major limitation of intelligent systems targeted for precision agriculture, search and rescue, and environment monitoring. New approaches to data association may be necessary to overcome the variation in appearance of natural environments.

This dissertation presents success with a map–centric approach, which builds on 3D vision to achieve visual data association across seasons. It first presents the new, Symphony Lake Dataset, which consists of fortnightly visual surveys of a 1.3 km lakeshore captured from an autonomous surface vehicle over three years. It then establishes dense correspondence as a technique to both provide robust visual data association and to eliminate the variation in viewpoint between surveys. Given a consistent map and localized poses, visual data association across seasons is achieved with the integration of map point priors and geometric constraints within the dense correspondence image alignment optimization. This algorithm is called Reprojection Flow.

This dissertation presents the first work to see through the variation in appearance across seasons in a natural environment using map point priors and localized poses. The variation in appearance had a minimized effect on dense correspondence when

anchored by accurate map points. Up to 37 surveys were transformed into year–long time–lapses at the scenes where their maps were consistent. This indicates that, at a time when frequent advancements are made towards robust visual data association, the spatial information in a map may be able to close the distance where hard cases have persisted between observations.

CHAPTER I

INTRODUCTION

A seeing, mobile machine that captures visual surveys of a natural environment can use the spatial information in a 3D map and 6D poses to see through the variation in appearance and achieve time–lapses across seasons. Figure 1 shows an example. One survey collects images over the length of a natural environment, which may consist of hundreds of unique scenes. As multiple surveys are acquired, image sequences start to form through the time elapsed at each scene. A transformation from multiple visual surveys into time–lapses connects the surveys and manifests the time elapsed through a set of well–aligned images at each scene.

This age of autonomy has machines that traverse and scan large environments over and over, with computers able to store and process that information, and vision able to provide increasingly intelligent analyses. Streams of long image sequences capture large swaths of environments as they are documented and explored. Natural environments pervade in this deluge. This necessitates methods to make sense of all that information.

Advances in perception have brought machines closer to automating tasks in natural environments that require repeated observation and management. Strides have been made towards precision agriculture [17, 5], search and rescue along forest trails [46], and a number of different types of environment monitoring [90, 68, 133]. Grand challenges for seeing, mobile machines in these scenarios may include automatically spraying certain crops, notifying the authorities, or compiling a record of changes. Addressing these goals requires advancements in support of long–term autonomy. As advancements are made toward solving the challenges that natural

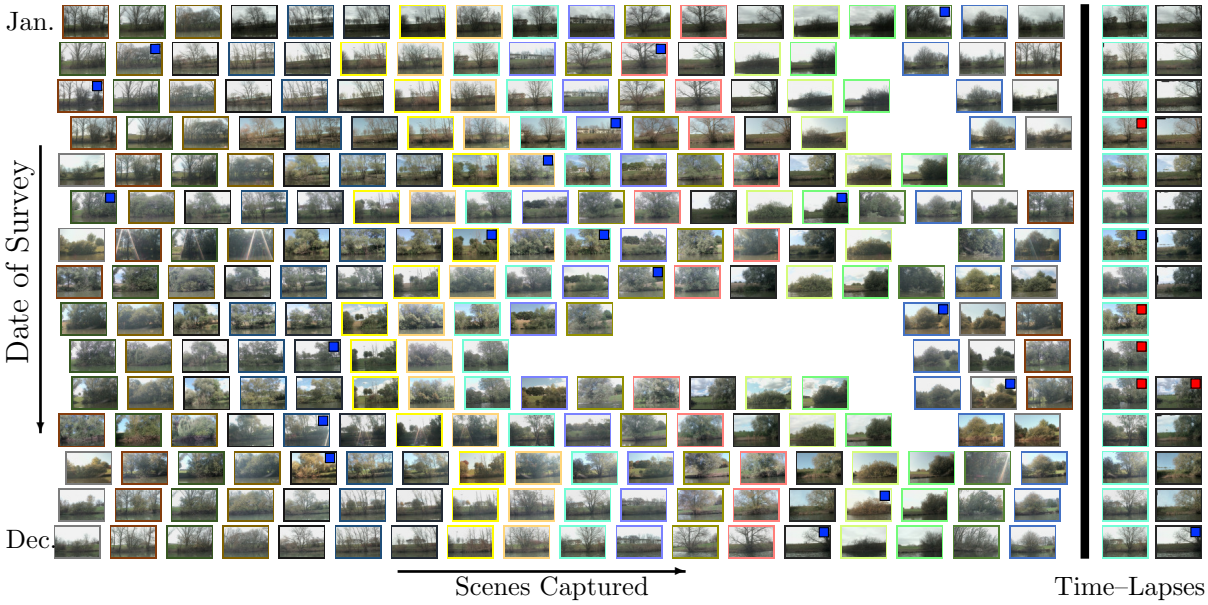

Figure 1: Depicting the transformation of unaligned, visual surveys of a natural environment into time–lapses. This example shows 15 of 37 total sessions from a year of surveys at 18 of 100 total scenes where time–lapses were produced. Each survey consisted of a video and the camera trajectory from a robot as it moved through a natural environment. Surveys from different dates were initially unaligned. The methods from this dissertation provided the ability to acquire loop closures across considerable variation in appearance, which was a part of a complete pipeline to transform the surveys into time–lapses. **left)** A hand–selected, reference image from a particular scene and survey was automatically found in the other surveys. The result set of images is shown bordered with the same color. There are 18 different sets. **right)** The images from two of the scenes aligned into time–lapses. Red squares denote misaligned images. Blue squares denote reference images.

environments present, more applications may become possible.

In some scenarios, the most useful information is in identifying precisely how an environment changed between observations. For instance, for a threat that may have camouflaged itself in some foliage, the best detection methods may require a comparison of before–and–after snapshots of the same scene. Fluctuations in forest fires and lava flows may require similar comparisons around their hazardous frontiers. More immediate assessment may be required in disaster response situations (e.g., tsunamis and earthquakes), in which affected structures in need of reinforcement (e.g., bridges, causeways, levees, dams) may need to be identified. A major challenge

of these scenarios is how to acquire visual data association of the same scenes over large spatio–temporal scales.

1.1 Challenges to visual monitoring using a mobile camera

Visual monitoring is traditionally the work of a stationary camera, which may be unsuitable for an expansive environment monitoring task. A single camera is limited to a single viewpoint. The area of coverage may be extendable using a pan-tilt actuator, but the field of view is still limited to the area that is visible from the camera's static location. A continuous stream of images of the same viewpoint would probably be unnecessary to capture the changes in an environment over a longer time–scale. The lack of change in viewpoint would simplify monitoring, however, because data association would not be required between images.

A mobile camera (hand–held, mounted on a vehicle, or on a robot), in contrast, can capture multiple scenes of an environment, but snapshots of the same scenes may no longer be captured from the same viewpoint. Without positioning that is as accurate as a stationary tripod mount, images of the same scene would have variation in viewpoint. Even if the camera is mounted on a train (which is on a track), snapshots from different traversals may be at slightly different offsets along the track. In mobile applications, the camera is typically more off-position than that.

Images with variation in viewpoint between them would have lost the implicit dense correspondence of images captured from the same viewpoint, which would make two images more difficult to compare. A natural environment may consist of complex scenes, with different things at different depths, which have different projections onto images at different viewpoints. The different projections would appear as differences in scene orientation, scale, and occlusions. Before two images could be directly compared, the one from a different survey that captures the same scene (preferably in the most similar way) to a reference image would have to be found. Two images could

then be compared where their pixels are found to correspond to one another.

Visual *data association* is the computational process which recovers corresponding pixels between two different images. Correspondences may be shared salient visual landmarks. A collection of matched landmarks at individual pixels or over several pixels would define correspondences for a subset of an image pair. The *dense correspondence* of two images is, in contrast, one that is specified at every pixel between two images. Monitoring tasks (i.e., from a stationary camera) typically seek to reason from an aligned image pair (whose dense correspondence is implicit).

The dense correspondence of two images may be possible to compute from a sparse set of correspondences. An accurate dense correspondence of planar surfaces is specified by a homography, which can be computed from four corresponding points. This is effective across images of low–laying land from a satellite, for example, whose birds-eye views capture relatively planar scenes. A homography is also accurate if the change in viewpoint is only a rotation. A homography would only provide, however, an approximation of the correct alignment between images of non-planar surfaces, which are characteristic of generic scenes in natural environments. A more complex alignment function, whose definition would need more than four corresponding points, may be required to align images from surveys of a natural environment.

1.2 Challenges to visual data association between images of a natural environment

Visual data association that strongly relies on appearance to find any correspondences may likely be unsuccessful between images of a natural environment. Dynamic processes of natural environments may limit the effectiveness of appearance in associating scene contents. The strength of illumination, the locations of shadows, and the type of weather vary between each survey. They sometimes vary within the same survey. Over longer time scales, seasonal changes in foliage may manifest as differences in size, shape, and color.

Images captured in outdoor and natural environments accumulate many other kinds of variation in appearance, which are typically lacking in indoor environments. On rainy days, water droplets diffract parts of the scene. On sunny days, sun glare and dust spots add noise. During spring, bugs and pollen sometimes land in front of the camera lens. Although extra precautions may be taken to avoid these issues (e.g., by capturing surveys on fair weather days), they are not completely unavoidable, especially in the wake of a disaster.

Natural environments with water make data association more difficult. Specular reflections due to water can lead to matching ambiguities between large sections of images. Water may be possible to detect and ignore at particular locations using information of the scene context (e.g., a beach), wind ripples, or changes in reflectivity. But small puddles and other reflective surfaces without those features could be more ambiguous. As the illumination conditions and the water quality change, so too does the degree of reflection.

Water can also be a major source of occlusion. Flooded areas may be indiscernible, aside from trees poking out of the water. After the water recedes, scenes may be coated with mud and debris. A monitoring task near the shoreline of a lake may project this kind of change more frequently as the water level fluctuates with the amount of rainfall. There, several meters of features may appear and disappear between surveys.

Dynamic processes of natural environments also add variation to the viewpoints from which an environment is captured. Debris after a storm may obstruct the usual path. As the water level of a lake fluctuates, the perspective of a scene could covary with it, e.g., if surveys are captured using an autonomous surface vehicle. Territorial animals (e.g., swans) may interfere with the robot's route in unpredictable ways. In cases like these, the same scene may only be viewed across a large perspective difference.

Reference Image | Unaligned Image | Target Alignment | Appearance–Based Alignment

Figure 2: Odd behavior can be observed when dense correspondence is appearance–based. Here tree branches were warped into foliage. The target alignment retained, in contrast, scene structure; the foreground trees were aligned despite their significant variation in appearance.

Cumulatively, the variation in appearance of a natural environment may represent a formidable challenge to visual data association. If there were only a few known factors of variation, a data association pipeline may be able to account for each one (e.g., factor out variation before data association, or as the distribution of data used to train a neural network). In a natural environment, however, there may be several factors of variation in appearance, including some that may be unaccounted for (e.g., a day of heavy pollen).

1.3 Hypotheses

An approach to dense correspondence that is guided by more than the appearance may be more suited to visual data association in a natural environment. Across seasons, for example, several trees could appear leafy with a bright skyline in summer, yet bare with the sky shining through them in winter (as in, e.g., Fig. 2). The winter image may best match the *appearance* of the summer image if the tree branches are warped into foliage. As humans we know, however, this is incorrect. We have experienced many trees (in the ways our sensorimotor repertoire allowed). Consequently, we know what a tree is, that it has a particular 3D structure, that it may abscise its leaves in fall, etc. The true correspondence for these images would align the branches from one to parts of the dense foliage of the other, leaving the tree structures intact.

One of the best natural neural networks for data association across seasons diverged through evolution from the primary reliance on appearance towards the primary reliance on spatial information, which suggests that a seeing, mobile machine may best solve data association in a natural environment using the spatial information (rather than the appearance information) from vision. Clark's nutcrackers cache over 30,000 whitebark pine seeds in autumn, which sustain them through winter, spring, and summer [71, 164]. An average of 1-15 seeds are stored per cache [164], which are spread out over a large geographical area (rather than clustered) [28]. Nutcrackers have an uncanny spatial memory to achieve this feat [89]. They integrate spatial information (before shape and appearance) of environmental cues like trees, rocks, and logs to find caches, including those that are out in the open or buried under snow [149, 24] (also see [50] for a nice video).

Identifying accurate correspondences in a natural environment may not always depend, therefore, on the appearance or the semantic correspondence of things; 3D landmarks and 6D localized poses may be enough. Trees keep the same position between observations, as do rocks, logs, the landscape, and many other objects that lack agency. The positions of things can, independently of appearance, indicate correspondences between images. A localized observer may be able to exploit this information to acquire fine–grained correspondences of things that appear substantially different. In other words, a seeing, mobile machine may be able to use 3D scene information to perform data association across seasons.

1.4 Research Questions

The primary research question is: how can a seeing, mobile machine acquire the dense correspondence of images of a scene of a natural environment if they bear little resemblance to one another? The hypothesis of this thesis is that accurate 3D scene structure can facilitate the dense correspondence of images. Algorithms already exist

for computing the dense correspondence of images of the same scenes. Algorithms also exist for recovering the 3D structure of an environment. This thesis is about recovering that structure and showing that it can provide a basis for data association across seasons.

The research hypotheses are addressed in six questions, the first three to identify tools with which the research hypothesis may start to be addressed, and the last three to discover how environment structure may be able to provide a basis for data association across seasons.

1.4.1 Question 1: With what dataset can data association across seasons be evaluated?

As part of this work towards answering these research hypotheses, the collection of a new dataset is described in Chapter 3. Before this dataset, few extensively captured a large natural environment over a long period of time. Several captured an urban environment from a vehicle. One captured four seasons of a natural environment in four videos from a train along a 700 km track in Norway. This dissertation presents a dataset of bi-weekly visual surveys of a 1.3 km lakeshore over three years, which is representative of an environment monitoring application. This dataset serves as the primary dataset for evaluating the work in this thesis.

1.4.2 Question 2: How can the 3D structure of a natural environment be recovered?

Existing work on structure from motion (SfM) and simultaneous localization and mapping (SLAM) have provided well–established frameworks for recovering 3D environment structure and 6D localized poses from image sequences, which is information that, once acquired, could be combined with dense correspondence to facilitate data association. These techniques often involve extracting and tracking keypoints across image sequences, triangulating their 3D positions, and fusing and refining the estimates of 6D camera poses and 3D landmarks using non-linear optimization. An

approach to SLAM is typically formulated around the particular constraints of the application. For environment surveying, recovering the 3D structure of the environment in order to perform data association across surveys is desirable, whereas real–time operation may be unnecessary.

An approach to visual SLAM was shown to recover the structure of an environment in consistent maps and poses for each survey in Chapter 4, but no loop closures were yet sought between surveys. Two approaches to building a map were compared. One was based on the use of SIFT features, which were sought to evaluate the use of a single consistent map between all the surveys. The other was based on the use of Kanade–Lucas–Tomasi (KLT) feature tracks, which provided one map for each survey. Results showed the former neither provided adequate environment coverage nor adequate matching capability across surveys. The latter provided, in contrast, the desired environment coverage and accuracy in the resulting 3D map. It left unclear, however, the question of how to fuse the structures from multiple surveys of a natural environment.

1.4.3 Question 3: How can images of the same scenes be found between images of a natural environment?

Finding images of the same scenes is a first step in performing dense correspondence or acquiring constraints between a pair of images from different surveys. It is unclear, however, how best to identify images of the same scenes of a natural environment. Finding an image of the same scene to a reference image is known as *image retrieval*. Rudimentary sensors (GPS and compass) may provide one estimate, while techniques for image descriptor matching may provide another. An answer to this research question is the first step in acquiring constraints between surveys.

Image retrieval was addressed in Chapter 5, which compared three wholly different image descriptors within a framework for image sequence alignment. The

comparison was between a dense correspondence approach, local image feature descriptors, and descriptors from one layer of a well–established neural network. The three appearance–based methods were also compared to a pose–based retrieval, which identified the nearest image according to rudimentary sensor values. Human labels provided the ground truth.

In addition to showing that dense correspondence was the most suitable descriptor for appearance–based image retrieval in a natural environment, its limits in retrieval were also identified. The dense correspondence approach lost matching power after approx. three months had elapsed between surveys. The pose–based approaches were most accurate, and applied well throughout the year. They were typically matched in accuracy by dense correspondence up to three months. The error in image retrieval climbed with the elapsed time between surveys, which reached a maximum error at six months, when the variation in appearance between surveys was greatest. At one year, however, the variation in appearance had diminished. Thus, appearance–based image retrieval was most accurate when applied between images from the same season.

1.4.4 Question 4: How well can an existing dense correspondence method apply to surveys of a natural environment?

Existing work has created algorithms to compute the dense correspondence between image pairs of similar scenes, which indicates that methods may already exist for aligning image pairs between surveys of a natural environment. Image pairs can be aligned if they are from the same scene category, which are those of the same scene with a different appearance or of different scenes with a similar appearance. The ability to compute dense correspondence between them indicates that those approaches to data association may be able to short significant variation in appearance. Indeed, some results have been shown between images of a natural environment. Yet, a specific study is lacking on the use of dense correspondence to align images across the variation in appearance of a natural environment for monitoring.

An evaluation of existing dense correspondence algorithms on a natural environment monitoring task is presented in Chapter 6, which showed that although variation in viewpoint can be eliminated in many images, there is indeed still a significant gap to be addressed before every image pair between surveys aligns well. First, a human was given the task of finding a single corresponding point between image pairs of the same scenes of a natural environment. The task was timed to demonstrate its difficulty. Then after three algorithms for dense correspondence were identified and compared, image pairs of the same scenes across multiple surveys were aligned. With the large number of images a human labeled as well–aligned, dense correspondence was specifically shown to provide accurate visual data association across significant variation in appearance. Aligned images were also shown to facilitate the manual detection of some changes. Yet, in finding that many image pairs did not align well, a significant gap was shown to exist in the ability to compute the dense correspondence of images of a natural environment.

1.4.5 Question 5: How can the 3D structure of an environment improve dense correspondence?

The 3D scene structure provides an untapped source of information in the visual data association computation, yet it is unclear how it can be used for dense correspondence across seasons. Although image retrieval may identify two images of the same scene (Chapter 5), without more information to guide visual data association, the appearance may be too different for two images of the same scene to align well (Chapter 6). A map and localized poses is potentially one source of guiding information. The 3D scene structure can be projected onto 6D localized poses to identify corresponding points between two images. Reprojected map points used in that way would provide position–based—independent of appearance—correspondences, and would thus be unaffected by the variation in appearance between surveys. Yet, it is unclear how that information could be used to improve visual data association.

Deficiencies in the matching power of dense correspondence were addressed in Chapter 7, which resulted in, in several cases, visual data association across seasons. Three different matching constraints were formulated to improve dense correspondence: 1) matching was pulled into forward–reverse consistency; 2) matching was pulled towards epipolar lines; and 3) map points defined position–based correspondence anchors. The first and the second were used to acquire one consistent map for multiple surveys across all four seasons. Given one map for all the surveys, visual data association across seasons was demonstrated. Importantly, the inability to align images across seasons was mitigated when dense correspondence was anchored using map points. As a result, visual data association had significantly improved where map point anchors were accurate.

1.4.6 Question 6: How can the techniques that established visual data association across seasons be expanded and better generalized?

Chapter 7 demonstrated significant results over variation in appearance of a natural environment on a small scale, leaving the question of its broader, practical applicability yet to be addressed. The application of map point anchors as priors for dense correspondence is limited primarily by map consistency. Acquiring one consistent map for many surveys is challenging if there are many incorrect loop closures. A primary challenge is, therefore, how to acquire robust loop closures for building a consistent map of a natural environment over a year. A successful broader application would indicate an advancement in the domain of environment monitoring.

Chapter 8 presents success with a map–centric approach, which was used to transform multiple visual surveys into time–lapses. A foundation of map point priors and geometric constraints were used within a dense correspondence image alignment optimization to align images and acquire loop closures between surveys. This framework produced many loop closures between sessions. Outlier loop closures were filtered in the frontend and in the backend to improve robustness. The evaluation showed that

year-long time-lapses were acquired for many different scenes.

1.5 Contributions

Four main contributions were made towards answering the research questions:

- A dataset was acquired and disseminated to support research towards long–term autonomy in natural environments.
- Dense correspondence was shown to be a primary method for visual data association in natural environments. The method was shown to outperform two other well–established techniques in an image retrieval task. It also provided a reliable number of loop closures between surveys with which a map for a year of surveys was acquired.
- A consistent map was shown for the first time to be a foundation for robust visual data association across seasons in a natural environment. The variation in appearance of a natural environment had a minimized effect on dense correspondence when anchored by accurate map points.
- Time–lapses were extracted at many locations along a natural environment, which captured its variation in appearance over many surveys in a year. An advancement is thus made in the domain of visual environment monitoring.

1.6 Overview

Related work in monitoring, data association, and dense correspondence is presented in Chapter 2. Chapter 3 describes the robot, the natural environment, and the dataset that was used to evaluate the work. Chapters 4, 5, 6, 7, and 8 address questions 2–6, in turn. The conclusions, limitations, and future work are part of Chapter 9.

CHAPTER II

RELATED WORK

A large body of related work has provided a foundation for the work in this dissertation, which are summarized here. Because this dissertation addresses monitoring in a lakeshore environment using an autonomous surface vehicle (ASV), which results in time–lapses of scenes from the environment, related work on monitoring are addressed first (Section 2.1). Extensive coverage of robust visual data association follows (Section 2.2).

2.1 Monitoring

Related work on natural environment monitoring is split into sections on building time–lapses (Section 2.1.1) and autonomous lakeshore monitoring (Section 2.1.2).

2.1.1 Building Time–Lapses

A number of recent approaches focused on building time–lapses in a natural environment from multiple viewpoints, which is related work that is highly similar to this dissertation. Dong *et al.* [34] acquire a dense point cloud from each session and then align them into a 4D point cloud for precision agriculture of a peanut farm. Loop–closures are acquired by applying a homography to find SIFT feature correspondences (time interval $<$ 1 week between sessions). Milford *et al.* [117] apply SeqSLAM to align images from multiple image sequences of a natural environment. Image pairs are aligned by applying an affine transformation to the correspondences obtained using an adapted SeqSLAM approach. Like Milford *et al.* [117], the methods in this dissertation align environment–long sequences of surveys, yet it is map–centric like that of Dong *et al.* [34], which provides much of the robustness to variation in appearance.

Publicly available photos of popular landmarks also capture a representative set for a transform into time–lapses. Techniques for large–scale scene reconstruction from mined internet photos adapt well into time–lapses as the reconstruction is temporally ordered. Martin-Brualla *et al.* [111, 110] reconstruct scenes into time–lapses by building a depth map for each viewpoint at each instance in time. A color profile is computed for each 3D track, from which the scene is reconstructed into a time–lapse. Zhou *et al.* [175] maximize correspondence consistency among the mesh of correspondences, or 'flowweb', of an image collection to align them. This dissertation also aims to make time–lapses whose images are more closely aligned, and it uses 3D structure build them.

2.1.2 Autonomous Lakeshore Monitoring

This dissertation explores autonomous lakeshore monitoring as an application area for a robust dense correspondence system, an area under study by multiple groups for many different reasons. Autonomous lakeshore monitoring can, for example, help improve water quality [70, 127], capture the environmental effects of dams [145], and survey rare plants [132]. The contributions of this dissertation can impact other monitoring tasks as well, including surveying abandoned mines [162], monitoring the frontier of forest fires [152], modeling the seafloor [79], and mapping and classifying farmland [13].

Mapping has been one of the primary components for monitoring. A few different SLAM systems have been established specifically for lakeshores. Heidarsson and Sukhatme [66] and Subramanian *et al.* [156] demonstrate SLAM on a lake from an ASV. In case a robot is repeatedly deployed on the same lake, Hitz *et al.* [69] show that 3D laser scans of a shoreline can be used to identify some types of changes. Their system distinguished the dynamic leaves from the static trunk of a willow tree in two different surveys collected in the fall and the spring. Dense laser scans can

also be used to ascertain crop growth [17]. Rather than use an ASV, Jain *et al* [74] demonstrated the advantages of using a drone for mapping a shoreline, which flies above debris and below dense tree cover.

The simple need to scale some monitoring tasks to the large spatial and temporal extent of lakes has motivated some solutions based on deploying ASVs. An ASV's sensor suite can facilitate monitoring, which can typically include the manual overhead of using many gadgets and sometimes awkward mobility on water. Tokekar *et al.* [163] track invasive carp using a technique involving radio tags. An ASV moves back and forth around popular carp spots of a lake while its antenna senses signals from tagged fish. Toxic cyanobacteria present another serious intrusion to lake ecosystems, which Hitz *et al.* [68] monitor using an underwater probe. An automatic winch on their ASV adaptively raises and lowers a probe to different depths in order to find the layer where cyanobacteria are concentrated.

2.2 Robust Visual Data Association

Methods towards robust visual data association in outdoor environments address the question of how to overcome variation in appearance between observations. Varying degrees of variation are captured in different visual data association tasks, which have led to many different solutions [105]. The related work surveyed here describes areas of research that have provided a foundation for the direction taken in this dissertation. Although this dissertation has not employed deep learning solutions, they are included within this context below, explored in more depth in Section 2.2.5, and are a main direction for future work (Chapter 9). In this section, related work is organized into six areas: Section 2.2.1) the use of hand–engineered visual features; Section 2.2.2) the use of features from a neural network; Section 2.2.3) image sequence matching; Section 2.2.4) image modification; Section 2.2.5) dense correspondence; and Section 2.2.6) exploiting prior knowledge.

2.2.1 The Use of Hand–Engineered Visual Features

Many different hand–engineered local image features have been proposed to address particular aspects of the correspondence problem. The success of SIFT features [103] has given rise to other local features, including SURF [103], ORB [143], BRIEF [15], BRISK [98], and more. Each defines a detector and a descriptor. The detector is meant to identify salient image locations. The descriptor is meant as an identifier of a specific location, which is used for matching. Both are applied together for repeatable feature matching that strives for robustness to changes in scale, viewpoint, and appearance condition, while also being computationally efficient. The different methods make different tradeoffs between degrees of robustness and speed.

Local image features are reliable for data association in some outdoor applications. Using SIFT features is popular for finding correspondences in images of urban environments (e.g., [87, 6, 65]). Glover *et al.* [48] achieve localization at various times of the day by integrating probabilistic data association based on illumination–invariant features with filtering. Their approach can map an outdoor environment in a way that is somewhat robust to scene variation. It does, however, generate many new descriptors for revisited locations. Because the features captured at night often appear entirely different, Johns and Yang [78] propose fusing features from day and night into co-occurrence maps, which allow for localization at any time of the day.

Unfortunately, SIFT matching (using OpenCV [10]) in natural environments often fails or returns too few features (less than 10) due to low contrast, intra-image similarity of the descriptors, and seasonal changes [56, 167], which has led to studies on the use of other image features. Krajnik *et al.* [88] evaluated many different local image features for use in natural environments. There, the BRIEF feature descriptor (and the variant GRIEF) outperformed SIFT and other local image features in a localization task across seasons with images that lacked translations. Prior work has shown that, among hand–engineered local image features, BRIEF, ORB, SIFT, and

SURF may perform similarly on images from a natural environment [59].

Several papers transitioned to whole–image matching to gain robustness to variation in appearance. Extracting descriptors for whole images has the advantage that salient image regions do not have to be identified beforehand. Performing data association at the granularity of whole images may, however, result in a loss of precision, but descriptors can be used for place recognition. Scenes have a spatial layout, which can be captured in the gist of the scene [129]. As-is, however, these approaches may have limited robustness to changes in viewpoint.

Image patch matching is an alternative to whole image matching, which retains some robustness to variation in appearance yet is more robust to changes in viewpoint. McManus *et al.* [114] utilize patches of images, called 'scene signatures', which are matched using classifiers and capture particular structures of each scene. Unfortunately, these approaches are not designed to produce pixel–level correspondence between matched image patches.

2.2.2 The Use of Features from a Neural Network

Current methods for state-of-the-art results in many areas of computer vision use neural networks, including solutions for robust visual data association. In the context of robust visual data association outdoors, Sunderhauf *et al.* [159] showed that condition– and viewpoint–invariant place recognition can be achieved using visual features from a CNN. Sunderhauf *et al.* [158] demonstrated that CNNs trained for object recognition can be used for whole image place recognition. They outperformed other techniques (e.g., SeqSLAM [119]) specifically designed for robustness to changes in appearance [159]. The descriptor provided by the third layer of a CNN provided the right balance of specificity and generality for scene recognition across seasons.

Several techniques have improved upon learned image patch matching for place recognition. Selecting a better set of training data is one of the best ways to improve

the performance. The method performs 10% better over many different datasets using a CNN specifically trained for place recognition [22]. Although a CNN trained for general place recognition may under–perform on scenes of a natural environment (see Chapter 5), a neural network that is specifically designed for and trained on images from a natural environment can acquire much better invariance to the conditions of its scenes [49, 102, 128].

Although Chapter 5 shows that a hand-engineered dense correspondence technique outperformed a CNN for place recognition (like that used by Sunderhauf *et al.* [159]), Section 2.2.5 points out how learning approaches can significantly improve the approach of this dissertation.

2.2.3 Image Sequence Matching

Several approaches are tailored for matching a sequence of images in order to find a single corresponding place. The use of multiple images can add robustness to variation in appearance where single images may be hard to match. This may be particularly useful in GPS–denied environments. Sequences of image templates can be matched directly [116, 4], paired up in a network flow [122], or as nodes of the data association graph [167]. Naseer *et al.* [122] showed that image sequences can be matched as a solution to the network flow problem through a cost matrix of matched descriptors.

Image sequence matching has alternatively been addressed as synchronizing image sequences—that is, finding correspondence at the level of the complete video instead of image-to-image. The matching solution is built consistently over the full sequence, which takes advantage of the frames with salient features to constrain the matches on less discriminative frames. These types of approaches may be especially relevant for time–aligning surveys of natural environments, which may have many uniform features.

Evangelidis *et al.* [41] synchronize two image sequences by maximizing the number of matching feature points between image pairs, and then refining the estimate based on image similarity. A database is made for the first sequence, which is queried with images from a new sequence to get a visually close frame. The result is refined using a 2D homography to find the best corresponding image. Based on their algorithm's performance to major changes in illumination between video sequences, they recommend a different approach if sequences have variation in appearance between them.

In a different approach to synchronization based on time–warping, Wang *et al.* [168] give a human the power to explore the space of possible synchronizations through an interactive system. The approach computes a matching cost for all potential pairwise matches using feature descriptors. The cost of matching a frame from the first sequence with a frame from the second sequence is inversely proportional to the number of matched feature descriptors between them. False positive feature matches and outliers are not filtered out because they have a relatively weak effect on the cost matrix. The sequence alignment algorithm uses Dijkstra's shortest path algorithm to extract the ridge line of best matching frames.

Naseer *et al.* [124] showed how Wang *et al.* [168]'s approach could be used for video sequences with seasonal variation. In this framework, many different approaches may be used for building the cost matrix through which two sequences are aligned. Naseer *et al.* [124] evaluated Histogram of Gaussians (HoG) feature descriptors and the node activations from a layer of a Convolutional Neural Network and found the latter to perform better.

The results of Chapter 5 show that a GPS and compass can provide comparable coarse matching accuracy across surveys for which the appearance is similar, and this level of accuracy can be maintained year–round [59].

2.2.4 Image Modification

Images may be modified to account for the difference in condition between two different surveys. A modification either removes a factor of variation from the appearance (Sec. 2.2.4.1), or copies the variation from one image to another (Sec. 2.2.4.2).

2.2.4.1 Removing Variation

Successes have been reported in filtering out specific types of scene variation. For example, explicitly representing light as a black–body illuminant allows for filtering out variation in appearance due to illumination (e.g., shadows) [25, 109]. Yet, the approach only works in the daytime, which leads to the awkward situation of using the resulting grayscale images in the day and unprocessed color images at night. Lowry and Milford [104] removed general factors of variation from images using a method based on PCA. The variation in appearance across a large training set of images is subtracted from the test set to reach the scene content that is invariant to the variation.

Long-term monitoring applications experience many other types of noise, which some have attempted to remove. Gu *et al.* [61] model an intermediate layer between the scene and the lens as one that attenuates or intensifies the desired image. Their approach filters out variation due to dirt and thin occluders, but fails for large occluders and sun glare. The limitations of approaches like this suggests that a framework based on them would not only be highly engineered, but could also risk failure in common situations.

2.2.4.2 Mimicking Variation

Perhaps the largest source of variation in appearance of natural environments comes from seasonal changes, which may be possible to copy into a target image. Laffront *et al.* [93] acquire a large set of aligned images with different scene attributes (e.g., sunny, fog, winter), which are manually labeled using crowdsourcing. The labels are

used to learn to estimate the magnitudes of the same attributes in new images and to learn transforms for appearance transfer. Rather than directly find correspondences across seasons, images are first transformed to appear like the same season [125]. This facilitates scene recognition. A major limitation of these approaches is the use of pre-aligned images to acquire a training dataset. Images have to be manually aligned, even for images captured using static cameras.

This dissertation avoids modifying the visual appearance in favor of relying on the scene geometry to gain robustness to variation in appearance.

2.2.5 Dense Correspondence

Methods for dense correspondence match every pixel across two images, which defines a transform between them, and which subsequently can make them suited to visual data association in a natural environment. In contrast to local image features or image patches, whole images capture the manifold structure of a scene [129], a pattern that may be more persistent across appearance change. In contrast to whole image matching, a dense correspondence also defines how one image transforms into another, which can make it less sensitive to changes in viewpoint. Related work here is organized by dense correspondence of images (Section 2.2.5.1) and of videos (Section 2.2.5.2).

2.2.5.1 Dense Correspondence of Images

A dense correspondence may exist between two images whether they capture the same scene or different scenes. SIFT Flow demonstrated the dense correspondence of two images by aligning whole images of SIFT features [100]. It can register images of different scenes that have a similar appearance, but it also works well for aligning images of the same scene that have significant variation in appearance. For example, Chapter 6 will show that the method is robust to some changes in appearance of a

natural environment. Scene structure provides the visual anchors through which spatial constraints pull the rest of the image into alignment. The method loses alignment accuracy, however, between images that are more than a few months apart.

Improvements to the methodology of SIFT Flow have built on the idea of matching whole images worth of point–based features [83, 170]. Instead of using image pyramids, Deformable Spatial Pyramids can be used to achieve scene correspondence much faster than SIFT Flow [83]. The method enforces spatial coherence by registering blocks of an image, rather than using spatial constraints between individual pixels. Because the finest layer of registration also lacks spatial constraints, the method is fast, but it loses precision.

Other techniques have shown improvements to SIFT Flow using different descriptors [84]. In case images have large degrees of rotation, image registration based on the Daisy Filter can be applied [170]. Yet, this and many other predominant image descriptors are hand-engineered, which has motivated [101] and [84] to investigate the use of CNN features for image registration. Although the dense correspondence optimization of SIFT Flow is improved by estimating an affine transformation in an alternate iteration with alignment, if SIFT Flow's optimization is used, features from a pretrained CNN outperform hand–engineered SIFT features [84]. In some cases, correspondence may be made more precise by matching generically spaced patches between images, rather than grid–sampled keypoints [62].

2.2.5.2 Dense Correspondence of Videos

The dense correspondence of videos can simplify the dense correspondence task as it allows for a few simplifying assumptions that reduce problem complexity. Video sequences captured while driving, for example, can be aligned by assuming the camera is only rotated between frames and then estimating a homography between images [32].

The alignments may not be exact, however, because the camera also often has a translation component. For video registration meant for more general applications, Sand and Teller [146] demonstrated an approach towards video matching that estimates a dense correspondence field using pixel matches and optical flow. Like a homography, however, the image warping function they compute is only an approximation of the true alignment function.

This dissertation builds on SIFT Flow to generate correspondences between surveys and then to acquire time–lapses. Chapter 6 gives a deeper introduction to SIFT Flow. Chapter 7 shows how it is built upon to achieve dense correspondence across seasons.

2.2.6 Exploiting Prior Knowledge

In addition to curating the data saved for localization (e.g., [97]), there is also significant work on using prior knowledge for improved data association rates. Data association has been successful to a prior 2D map [2], to one of multiple prior runs [23], or to a 3D model [174]. Churchill and Newman [23] showed that increased localization rates are possible if a new 'experience' of a scene is saved each time localization fails. Multiple experiences are acquired where scene change is more significant. Zhou *et al.* [174] train a neural network to infer the 3D model of an object given its query image, which is subsequently used to infer the correspondence between two images of the same object type. They showed that correspondence across significant change in appearance can be achieved if the 3D structure is known. Reprojection Flow (Section 7.1) is based on the same principle: reprojected 3D points can indicate how to anchor image alignment.

2.3 The Impact of Deep Learning on This Dissertation

Deep learning has become the dominant approach for addressing a large number of the problems addressed in this dissertation. Although each chapter describes results

Table 1: Datasets with images of a natural environment, which could be useful for training neural networks for image retrieval and dense correspondence.

Nordland	[153]	Four videos of a natural environment as seen from a train in four seasons.
SFU Mountain	[12]	Seven traversals of a woodland trail over a year.
FinnForest	[36]	Two videos of a wooded environment in summer and winter.
Devon Island	[44]	A 10 km traversal of a vegetation–free, rocky environment.
Outdoor Webcams	[73]	A global network of outdoor webcams, a subset of which capture natural environments.
iWildCAM	[7]	A large number of images from outdoor camera traps used to capture images of wildlife across seasons.
MAWI United GmbH	[112] (c.f., [45])	High-end CGI of multiple natural environments for UE4.

that led up to the framework in Chapter 8, many components are now significantly outperformed by learned ones. This section points out new learned methods, roughly organized in the same order as the body of this dissertation. Section 2.3.1 describes training data, Section 2.3.2 visual SLAM in natural environments, Section 2.3.3 image retrieval, and Section 2.3.4 dense correspondence.

2.3.1 Training Data

Because the availability of a large amount of training data is part of the story on the success of neural networks, in Table 1 I point out multiple datasets of natural environments, which can provide weakly and fully supervised training data.

In case there is not enough training data for a particular environment, it can be taken from varied sources that have been adapted to the target domain (i.e., domain adaptation [26]), as long as (in many cases) the final network has been fine-tuned from real images resembling those of the target environment. A CNN can perform significantly better if it is designed and trained specifically for the task it is evaluated

on [3]. However, the training process can utilize data from other sources to pre-train a neural network. After which, the weights can be fine–tuned (see e.g., [67]) to the target domain using a much smaller amount of data. This is highly useful for robotics applications, where data gathering may be prohibitively risky, time–consuming, and expensive [9]. Leveraging data from a different environment or a simulated one could help cover edge cases or make up for lacking types of variation.

Both pixel-wise and image-wise supervised training are possible using the images from the dataset described in Chapter 3 and the datasets in Table 1. Pixel-wise supervised training data (i.e., image pairs whose dense correspondence is given) is available due to a prior dense correspondence algorithm, from simulated environments, and from outdoor webcams. A transform can be applied to an image pair to misalign one before both are passed into the network [140]. Image-wise supervised training data (i.e., image pairs of the same or different places) is available on geotagged images, and images that we otherwise know captured the same scene [3].

2.3.2 Visual SLAM in Natural Environments

The map–centric approach to visual data association of this dissertation ultimately relies on the underlying accuracy of the map and the camera trajectory, for which impressive results using learning approaches have been achieved. State-of-the-art indirect methods are now based on the use of learned local image features [147, 161]. Rather than train a network to find salient points in an image (e.g., corners [30, 31]), the keypoints can also be found automatically [8, 138]. A network can learn to detect keypoints at pixels that are most discriminative for matching [138]. A network may produce a dense set of descriptors, with the detector identifying which ones to use for matching. The most discriminative descriptors are learned using the average precision loss [64, 138]. The best current approach to use for indirect mapping (as in Chapter 4) may be to use the R2D2 descriptor [138], which also detects a nearly uniform set of

keypoints.

A direct SLAM method that utilizes direct visual odometry may now be, however, the best approach for mapping a natural environment (e.g., the structure of DSO [39] in an optimization framework that supports loop-closure detection and correction with, e.g., iSAM2 [81]). Direct visual odometry methods on the Symphony Lake Dataset have reached high–fidelity maps [169, 20]. They use more of each image for frame-to-frame matching, which provides more accurate and more robust model parameter estimates. Although direct methods have typically been failure prone in cases with inaccurate initial estimates and variation in illumination and weather patterns, those limitations have recently been circumvented as well [166]. The GN-Net [166] converts an image into a feature space that facilitates camera relocalization in those conditions.

2.3.3 Image Retrieval

Although the combination of absolute pose plus a hand–engineered visual feature descriptor (described in Chapter 5) was effective for the image retrieval in this dissertation (i.e., images almost always captured the same scenes; see the number of misaligned images in Fig. 32), state-of-the-art image retrieval is learning–based. Learned image descriptors can identify images of the same scenes across significant variation in appearance and viewpoint [159, 3, 136, 137]. A neural network designed for image retrieval typically has a deep encoder, which ensures the last layer of features have the spatial support of the whole image. A pooling layer is added to the end of the network, which pools the extracted descriptors into a fixed image representation (see e.g., [3, 136]).

A network can be trained to perform image retrieval using images whose locations are known. The fact that images are known to have been captured at the same place or a different place is used as a supervision signal for training [3]. The image of the

same place that matches best should be embedded to a closer descriptor than all other far–away images [3]. That property is best captured using the average precision loss function (compared to the contrastive or the triplet loss functions) [137, 64]. The average precision is optimized when every positive image is ranked above all negative images.

A number of ideas have been proposed to enhance the accuracy of descriptors for image retrieval. Clearly, for tasks in natural environments, a network should be trained on images with a similar distribution of features [51, 128]. Concatenating multiple descriptors may bring an advantage if some images have few salient features [42]. A network for image retrieval can include a head for local image feature detection and description to help leverage the advantages of joint learning [147]. It is also possible to go a step denser and add a head for dense correspondence, which can improve the ranking of retrieved images [95]. In networks with local image descriptors, it may also be interesting to study how to rank images according to the covisibility of their local feature descriptors.

2.3.4 Dense Correspondence

Although some recent approaches have defined new optimization algorithms for dense correspondence [85, 139, 175], state-of-the-art methods are now learning–based [141, 115, 75]. A convolutional neural network can be trained to extract a feature for every pixel, which is used to perform dense correspondence. The output layer (or with a hierarchy of layers [120]) of an encoder has a resolution that is the same size as the image (after upsampling if needed), which defines a descriptor at every pixel. A dense correlation map (size HxWxHxW) is computed for every pixel to every other pixel, which is differentiable and amenable to end-to-end training. The flow can be taken as the e.g., neighborhood consensus [142] or after applying further constraints that improve the alignment [141].

The limited performance of appearance–based dense correspondence in this dissertation led to the utilization of multiple information sources (segmentation 6.3.2, reciprocal consistency 7.2.2, geometric constraints 7.2.3, correspondence priors 7.1.2), which have also been advantageous in learning–based methods that perform multi-task inference [21, 115, 141, 91, 174, 75]. Multi-task learning seeks to jointly infer different signals from the input data [94, 107]. Loss functions of each output can be framed using the outputs of different heads, which represents a consistency loss [175, 174]. That avoids the need to frame the loss around a ground truth value. An archetypical example is the joint inference of optical flow, semantic segmentation, instance segmentation, and relative pose of [107]. A network trained using a loss function that combines multiple outputs with a learned weighting (rather than a hand–tuned one) can outperform one with separate modules trained individually [82].

A different way to align images is to utilize dense depth maps. Given the known pose transform between two camera poses, a depth map provides a straightforward alignment of an image pair. The transform specified by the pose offset can, together with the depth at each point, be used to reconstruct the image at the same viewpoint as the reference frame. In case there are any inaccuracies in either the pose offset or the depth, and the appearance provides some useful information, dense correspondence (with a small hypothesis space) can be added to help complete the alignment. This idea is similar to the implementation of ICP-H [135], described in Section 8.5.2.

Depth can also provide extra guidance for the correct correspondences in a learning-based approach [94, 107]. A sparse depth map may come from visual SLAM. The sparse depth plus an RGB image can be extended into a full depth map using dense depth completion (see e.g., [171]). In case predicting a dense depth map is inaccurate and computationally demanding, as found by [166], the loss function may still incorporate sparse depth priors as guidance for correspondence.

2.4 Summary

This dissertation addresses the transformation of multiple visual surveys of a natural environment into time–lapses. It defines a map–centric approach for obtaining loop closures across challenging variation in appearance between sessions. It shows (in Chapter 5 and Chapter 6) that SIFT Flow could be used for data association across consecutive surveys; it worked best among several appearance–based methods for place recognition and its limit in appearance–based data association is around three months between surveys. Given that appearance–based data association using SIFT Flow can be effective up to three months, a search for loop closures is applied between pairs of surveys up to that time limit. Between pairs of surveys, a new pipeline is applied for visual data association, which utilizes Reprojection Flow and incorporates the scene structure (Chapter 4) within the dense correspondence optimization (Chapter 7). Time–lapses are produced by repeated dense correspondence across multiple surveys (Chapters 7 and 8).

Although this dissertation does not use a neural network for visual data association, it is complementary and unbiased to the particular appearance–based descriptor. (as long as the descriptor is used for dense correspondence). Higher data association accuracy could allow for longer periods of time between surveys. Across surveys where appearance–based data association sparsely spans the range of variation in appearance, where the training data is mismatched or is limited, where perceptual aliasing is high and the relative poses between surveys are accurate, or where verification with a map is desired, this dissertation will show that reprojected map points could provide anchors for visual data association.

CHAPTER III

SYMPHONY LAKE DATASET*

A growing homogenous space of publicly available robotic vision datasets capture a roadway from a car, which the release of Symphony Lake Dataset can help to diversify. Interest in creating autonomous driving vehicles has contributed to the growth and availability of roadway data. Work on perception has benefitted from the fact that these images are captured outdoors, and sometimes over long–term time periods (e.g., [108]). Yet, a roadway is highly structured, which could simplify perception and lead to non-general algorithms. A long–term dataset of a large–scale natural environment would add breadth to this space to advance research in perception.

Simultaneously, advancements in deep learning have generated interest in massive datasets. Baseline performance in tasks like scene classification improve with the amount and the diversity of the training data [172]. With millions of exemplars, some basic labeling tasks have reached nearly human–level performance, while some advanced game AI have surpassed the best humans. Advancements seem to come in parallel with the availability of data. The release of Symphony Lake Dataset may contribute to this growth in results for perception in natural environments.

This chapter describes the release of Symphony Lake Dataset, 121 visual surveys of the shore and the island of Symphony Lake in Metz, France. The 1.3 km shore was surveyed using a pan–tilt–zoom (PTZ) camera mounted on an unmanned surface vehicle (see Fig. 3). The camera faced starboard (or port for the island) as the boat moved in parallel with the shore. The boat was deployed on average every 10 days from Jan 6, 2014 to April 3, 2017. Over 5 million images were captured.

*This chapter is a paper that was published in the 2017 International Journal of Robotics Research (IJRR) [53].

Figure 3: The GTL Clearpath Kingfisher as it circled the perimeter of Symphony Lake.

The 600 GB dataset is released in two sets: 1) 4 GB full surveys and 2) 200 MB sub-sampled surveys. The surveys include GPS, IMU, and compass data, which is synchronized to the 704 × 480 @ 10 fps color images. Readings from the 2D LiDAR are also included. Each survey is available for individual download on a dedicated website at `http://tale.georgiatech-metz.fr/symphony/`.

This chapter uses a similar structure to [108] (with permission), which is an archetypal robotics dataset paper. Their autonomous–capable car captured a suburban neighborhood in Oxford, UK twice a week, on average, for over a year. The full view of street scenes around their vehicle is captured in 3D LiDAR and image data (among data from other sensors). In contrast, this chapter captures a natural environment week-to-week as it evolved over three years and the data consists primarily of side-view images.

3.1 Survey Setup

3.1.1 Platform

The robotic platform is the Kingfisher M200 unmanned surface vehicle (USV) from Clearpath Robotics (see Fig. 4). The USV has the style of a pontoon–boat. A 0.55 m × 0.80 m metal base connects the top of two 1.3 m-long pontoons. The back of

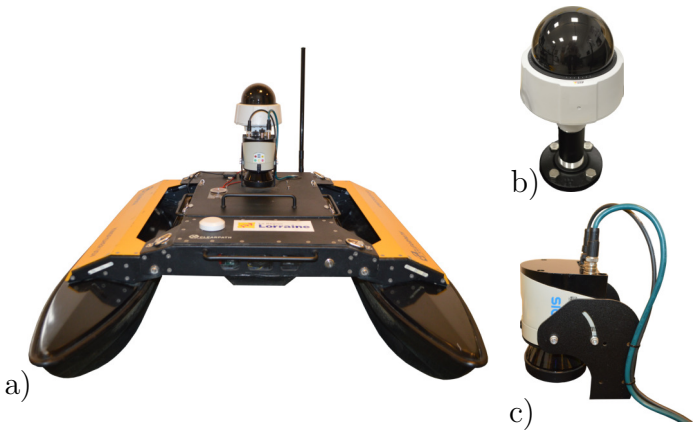

Figure 4: **a)** Front view of the GTL Clearpath Kingfisher and side views of **b)** the PTZ camera, and **c)** the 2D LiDAR. A survey is initiated using a computer connected via the wifi antennae in the back. As the motor in each pontoon propels the robot, the camera pans starboard (or port for the island). The laser range–finder measures the ranges to obstacles, which are used to maintain a 10m distance to the shore. The GPS, the compass, and the IMU measure trajectory values while the computer inside the waterproof compartment records all of it.

each pontoon houses a jet thruster, which propels the boat up to 1.7 m/s. A power differential between the motors turns it.

A 40 Ah nickel–metal hydride battery powers the USV. The battery is secured inside a compartment in the metal base before each survey. It has enough charge to move the boat at nearly 0.35 m/s for over an hour. While stationary it can power the sensors and the onboard computer for up to 10 hours.

The USV is equipped with four primary sensors:

PTZ Camera:	Axis P5512-E. 360Pan. 180Tilt. 12x zoom. 704x480 @ up to 60 Hz. 3.8mm Lens. 51.6HFoV.
2D LIDAR:	SICK LMS111. 20m Range. 0.5Resolution. 50 Hz. 270HFoV.
GPS:	U-Blox LEA-6. 5 Hz. 2.5m
IMU:	CHR-UM6. 2Pitch and roll accuracy. 5Yaw accuracy.

The metal base of the USV has a waterproof electronics bay inside it and a platform bay on top for the sensors. The GPS and the onboard computer are housed

within the electronics bay. The PTZ camera, the laser range–finder, and the IMU are mounted to the platform bay. The camera was mounted behind the laser, high enough for an unobstructed view. Because the laser range–finder is mounted facing forward, distances to objects behind the USV are not measured.

Sensors fed their data to an embedded computer. The computer has an Intel Atom Z530 CPU (1.6 GHz, 2 threads, 32-bit), 1 GB RAM, and a 16GB SanDisk SSD U100. In addition to planning the robot's motion using LiDAR data, the computer processed and stored sensor readings. The hard drive was large enough to store over a survey of data.

3.1.2 Symphony Lake

Symphony Lake is 2 km south east of Metz, France, across the street from GeorgiaTech–Lorraine (see Fig. 5a). It is approximately 400 m at its longest point and 200 m at its widest point. The total area of the lake and its surroundings spans 6 ha. It also has an 80 m-wide island in the middle. The lakeshore perimeter, including the island, is about 1.3 km.

The lake was created in 1986 to prevent floods in Metz. One main inlet and a single outlet control the flow of water down a creek. During periods of heavy rain the lake's water level can increase several meters. The bank of the lakeshore is fairly steep, which keeps the water contained in the basin.

The nature of the lakeshore is varied. Some areas are surrounded with shrubs, bushes, and 20 m-tall trees. There are areas with boulders, sand, and grass. Buildings loom in much of the background. They are closer to the shore on the north east side.

The land around the lake is used to promote recreation. The grass is periodically mowed, and the other flora are sometimes trimmed or removed. A 1.35 km fitness path and a nature trail encircle the lake. Fishing, tanning, biking, jogging, and walking are common.

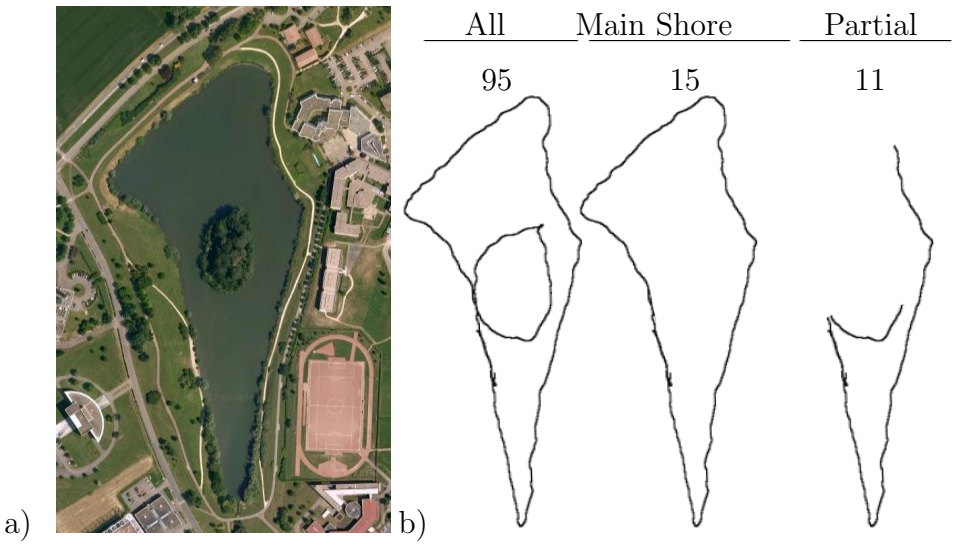

Figure 5: a) Symphony Lake from the perspective of Google Maps Satellite View, and b) depictions of the different trajectories of the robot with their number of occurrences. The boat circled approximately the entire perimeter (95 cases), missed the full island (15 cases), or otherwise partially traversed its route (11 cases). That is, *Main Shore* includes surveys with partial island coverage. *Partial* here is illustrated with one example of a partial route; each one was different.

3.1.3 Behavior

The trajectory of the boat on Symphony Lake is shown in Fig. 5b. The boat was typically deployed from the west side of the shore and was pulled out at the same location after one complete run. It was sometimes pulled out at other locations in order to reset automation or to end the survey. Surveys could be started anywhere along the shore.

The USV circled roughly all of the approx. 1.3 km lakeshore, which took it nearly 70 minutes. Surveys occasionally took longer (due to e.g. wind). Several surveys captured less of the perimeter. Fifteen captured all of the main shoreline and were ended without the full island. Eleven captured parts of the main shore and the island. Rain, battery charge, automation failures, and swan interference were typical limiting factors. A survey was sometimes cut short if multiple issues occurred.

A finite state machine generated the robot's trajectory. First, the USV navigates

to a position 10 m from the shore and the camera pans starboard. The USV maintains its 10 m distance as it circles the perimeter in the counter–clockwise direction. The main shore survey continues until the USV crosses a virtual transition line, which extends west from the GPS position of the island's center. The boat surveys the island after it aligns itself 10 m from the island's shore and the camera pans port. The same transition line is used to shift back to the main shore survey.

The boat's trajectory was replanned at a rate of 5 Hz. A local lattice planner with a 10 m horizon provided the set of behaviors to choose from. Each one was evaluated using ranges from the LiDAR. The planner chose smooth trajectories that also kept the USV 10 m from the shore and at least 2 m away from obstacles. If the USV got closer to obstacles, however, the planner diverted its course more abruptly to avoid collisions.

The boat was monitored for the duration it was deployed, except while it circled the island. A human intervened if necessary to keep the USV moving in the right direction or to completely reset automation. An automation failure sometimes occurred at the south end of the lake where the sharp turn caused the USV to oversteer and spin in place. The GPS position sometimes fluctuated enough that the transition to the island occurred at the wrong places. A human also intervened to avoid fishing lines and to correct the boat's path near swans.

3.2 Survey Data

3.2.1 Data Collection

Each survey consists of image, LiDAR, pose, and state data. There is one file per image, a set of files for the LiDAR readings, and one file of all the pose and state information. Thousands of images and LiDAR readings are saved per survey. Each 704×480 image is stored in a jpeg format with a slight compression. Each LiDAR reading provided 541 range measurements across the 270rc in front of the robot. New

readings were recorded at a rate of 50 Hz.

Readings from the other sensors are saved to an auxiliary file. Pose data includes the 2D position (m) from the GPS, the heading (deg) from the compass, and the angular velocity (deg/s) from the IMU. The auxiliary file has one set of pose data per line. Each line in the file corresponds to one image.

State information is saved with the pose data to guide data processing. The camera state includes its dynamic pan and tilt values, as well as its static intrinsic parameters. A pan value of approx. ± 1.57 identifies when the survey is occurring. Other values indicate transitions. A positive value indicates the USV is surveying the main shore, a negative value the island. Other information includes the time, the image number, the battery charge, and the RC controller state (i.e., whether the USV was operating in autonomous or manual control mode).

3.2.2 Survey Package

The dataset is packaged according to its size. Thus, 4 GB surveys are available for individual download rather than as one large chunk. A 20x downsampled, 200 MB version of each survey is also available, which targets use cases that require images with less overlap. The LiDAR data is made available in its own package. Using the May 2, 2014 survey as an example (referenced as 140502), the files for one survey are:

- *140502f.tar.gz*
  - Around 41,000 jpeg images in a hierarchical directory format with 1000 images per directory. Files are referenced using the value of an image counter. Images are numbered between *0000.jpg* to *0999.jpg* per directory, and directory names typically span approx. *0000/* to *0041/*.
  - *image_auxiliary.csv* - lines of image timestamp, image number, pose readings, and state information, in that order, with a new line of values every 0.1 s. The full list is shown in Table 2.

Table 2: The order of values of one line of image_auxiliary.csv.

1	timestamp	seconds	image time
2	image number		the image file index
3	UTM E	meters	GPS position in UTM 32 N
4	UTM N	meters	GPS position in UTM 32 N
5	compass	degrees	NED frame
6	camera pan	degrees	positive for starboard
7	camera tilt	degrees	
8	fx	pixels	
9	fy	pixels	
10	cx	pixels	
11	cy	pixels	
12	image width	pixels	
13	image height	pixels	
14	omega	degrees/second	IMU angular velocity
15	battery	voltage	
16	RC state	enum	1 - in range. 2 - in use.

- *140502d.tar.gz*
  - Contains the same files as *140502f.tar.gz*, except with 1/20 of the readings.
- *140502l.tar.gz* - a tar file of LiDAR data for a survey. LiDAR data is saved to a set of time–ordered *csv* files, each with a timestamp in the first column, followed by 541 range readings (in meters) in the following columns. Roughly 110 *csv* files per survey, each with around 1,800 scans, make a total of about $110 \times 1,800 = 200,000$ scans per survey.

To assist in the selection of surveys, a summary video for each survey is available on the GTL website. The summary is a subset of images, taken every 1.5 m of the USV's motion, compiled into a video.

3.2.3 Additional Files

Additional files are included in Symphony Lake Dataset that apply to all the survey data:

- *ParseSurvey* - C++ code to interface with the survey data. The code reads an

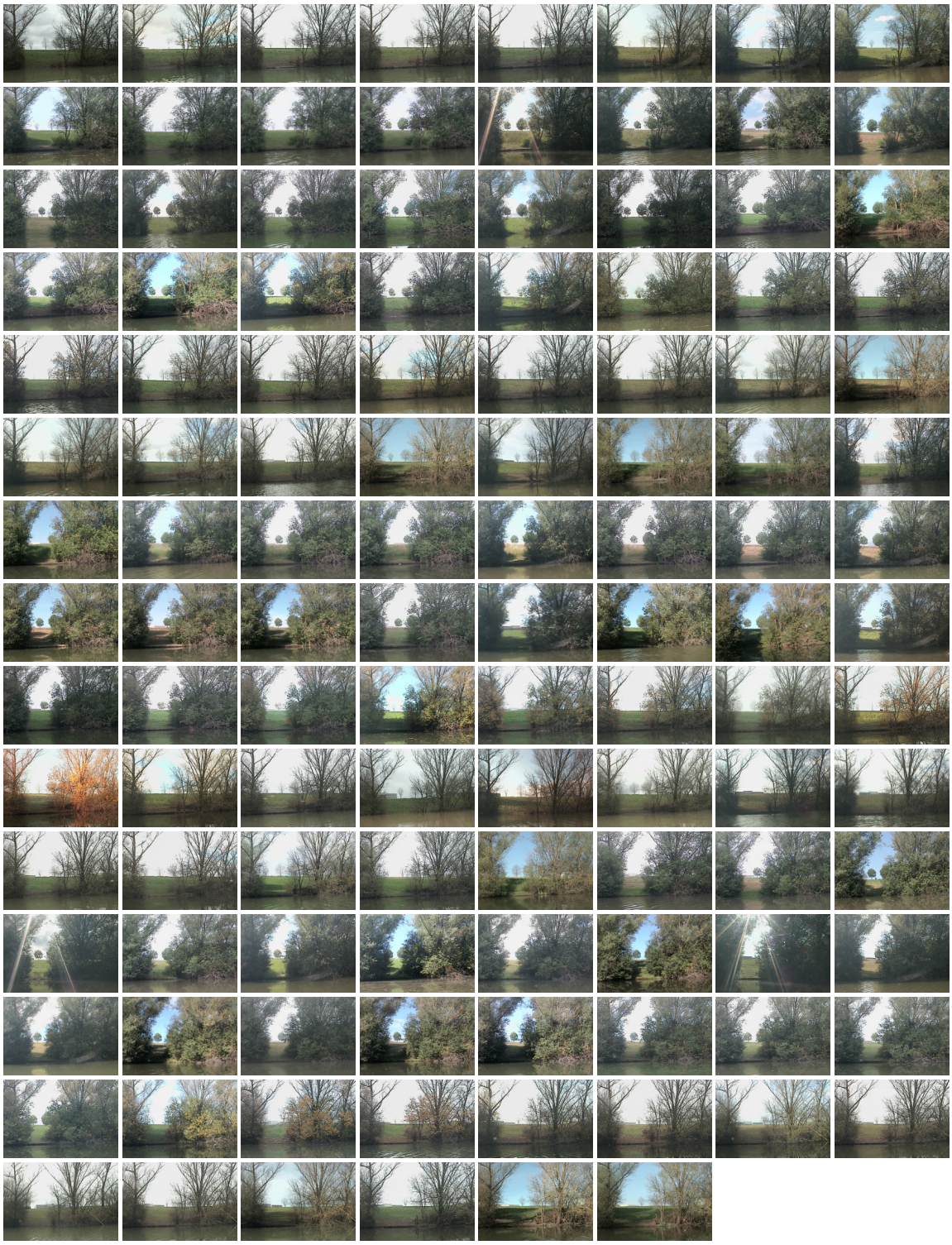

Figure 6: Montage of images of one scene of the lakeshore from 118 surveys, inspired by [108]. Consecutive surveys are in row–major order. This scene primarily has features from an unstructured environment, captured over three years. In the montage of [108], in contrast, the structured, street environment has some features whose appearance is more static (e.g., the sign post they used as a reference), which can simplify data association.

image_auxiliary.csv file and can provide the file paths to the images. It also converts raw sensor data into the camera pose.

- *camera_calibration.txt* - The full set of calibration values for the PTZ camera. A sequence of checkerboard images was used to obtain the calibration parameters.
- *sensor_positions.xls* - A spreadsheet of sensor positions for GPS, the PTZ camera, and the 2D LiDAR.
- *catalogue.xls* - A catalogue that collates survey attributes like those visible in Fig. 6. Each entry consists of survey duration, distance traveled, weather pattern, presence of noise, and more. The attributes for a survey were manually populated while viewing its summary video.

The following section characterizes the dataset using the catalogue.

3.3 Dataset Characteristics

A collection of 5,031,232 images from 121 visual surveys compose Symphony Lake Dataset. Figure 7 shows the timeline of surveys, which span from January 2014 to April 2017. A survey was captured on average about once every 10 days. Surveys were missed during weeks of heavy rains, if I was traveling, or if the lake was frozen.

The evolution of one scene across all three years is shown in Fig. 6. There are large changes in appearance. Different changes are more apparent across different time scales. Changes in weather, illumination, viewpoint, and water reflectivity are apparent in many comparisons week-to-week and year-to-year. Large changes in foliage become apparent season-to-season. The montage also shows some cases of noise (e.g., sun glare).

3.3.1 Perspective Differences

Although surveys often captured the same scene, perspective differences occurred due to the fact that the images are captured from a mobile robot. The camera trajectory

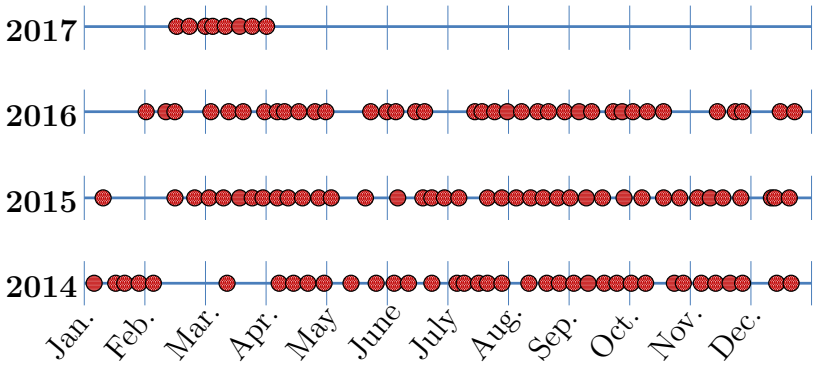

Figure 7: Timeline of surveys of Symphony Lake between Jan. 6, 2014 and Apr. 3, 2017. The lake was surveyed 37 times in 2014, 39 in 2015, 37 in 2016, and 8 (currently) in 2017.

was typically at least slightly different between surveys, while sometimes factors were present that contributed to more substantial variation in viewpoint, as shown in Fig. 8. For example, the entire trajectory changed in times of high water. Aside from the fact that the camera has a fixed height from the lake's surface, more water meant the boat moved more inland. Strong winds also skewed the boat's trajectory because power to the boat's motors was set to a constant value. Fortunately, in those cases the boat could still capture a survey automatically. Perspective differences also occurred when manual control was required.

The boat's trajectory was also effected during the variable amounts of time it was in the company of a swan. A pair of swans occupied Symphony Lake. They were always peaceful towards the boat. Often during nesting season (late March through early May) the male exhibited its dominance nearby. It learned how to divert the boat from its path (swim up a side of the boat), which it typically did near the island (the annual location of the swans' nest). On these occasions the boat was manually steered on its path, at a comfortable distance from the swan, but the boat sometimes veered off course to automatically avoid the swan if it was beyond the line of sight.

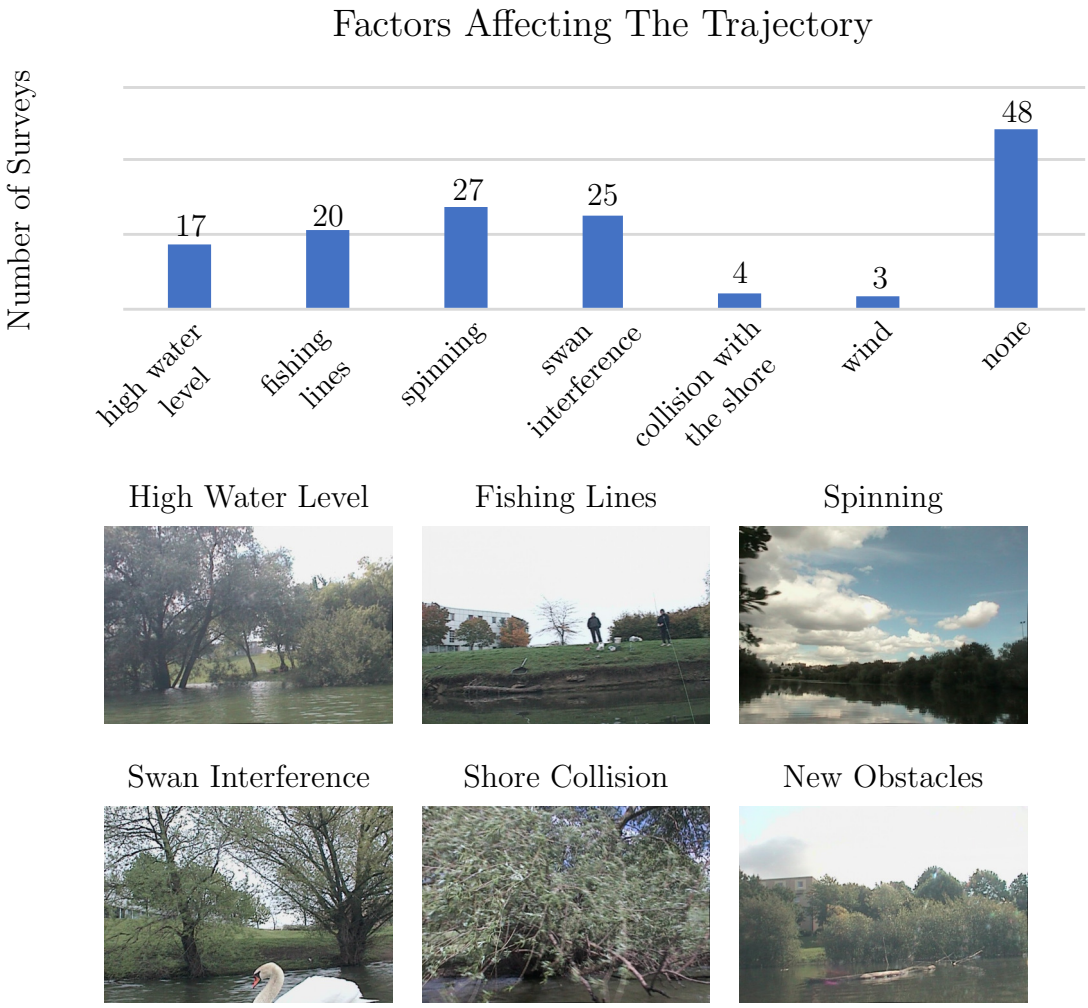

Figure 8: Factors affecting the boat's trajectory and their occurrence in the dataset. Although some factors were present throughout a survey (high water level, wind), others were localized to specific places (fishing lines, automation error, collisions, and new obstacles). A combination of these factors affected several surveys.

3.3.2 Variation in Appearance

The fact that images are captured outdoors adds to the variation in appearance caused by perspective differences. Illumination is, for example, non-uniform and varying, and a function of the sun's position in the sky and the particular weather pattern. The sun's position varied, in turn, with the time of the day and the day of the year. The

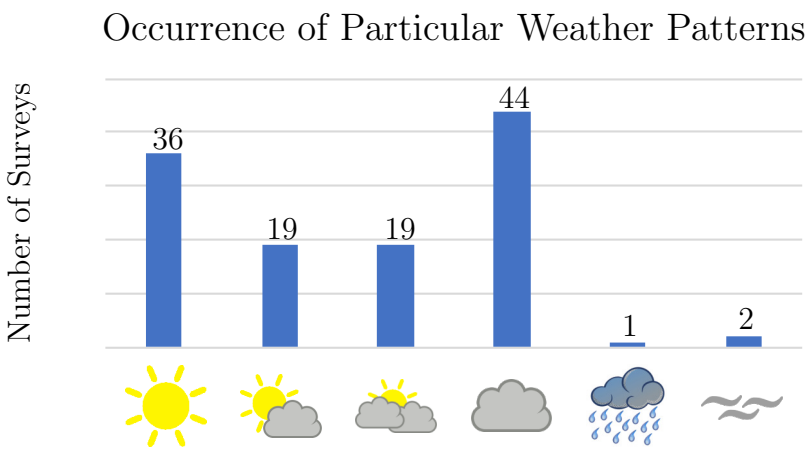

Figure 9: Occurrence of particular weather patterns in the dataset. The surveys spread well between sunny and overcast. In general, surveys on rainy days were avoided. Two surveys captured fog.

more sun, the stronger the illumination, yet the stronger the shadows. The more direct sun, the more intense the sun glare.

Figure 9 shows that the surveys varied well from sunny to overcast. Rainy days were avoided because raindrops on the dome of the PTZ camera blurred the images. Fog was captured twice. Snow could not be captured due to the nature of the dataset. There was seldom more than a little snow, which typically occurred with a frozen lake.

Variation in appearance in this dataset is perhaps stronger than in street datasets because surveys captured a natural environment. Most images captured flora, which changed significantly across seasons. In the winter the background can be seen through trees and bushes, but in the summer and the fall it is occluded. The structures of some plants are occluded by their own foliage, which makes their recognition and association across seasons difficult. Foliage also often lacks strong features, and resembles nearby plants.

Being on a lake means that the bottom 18% of each image captures water, which varies from murky, to wavy, to reflective. Although a comparison of two images may be a success if the water is disregarded, it can interfere with the process. The flora, the shoreline, and the water blend together on days when the water is reflective.

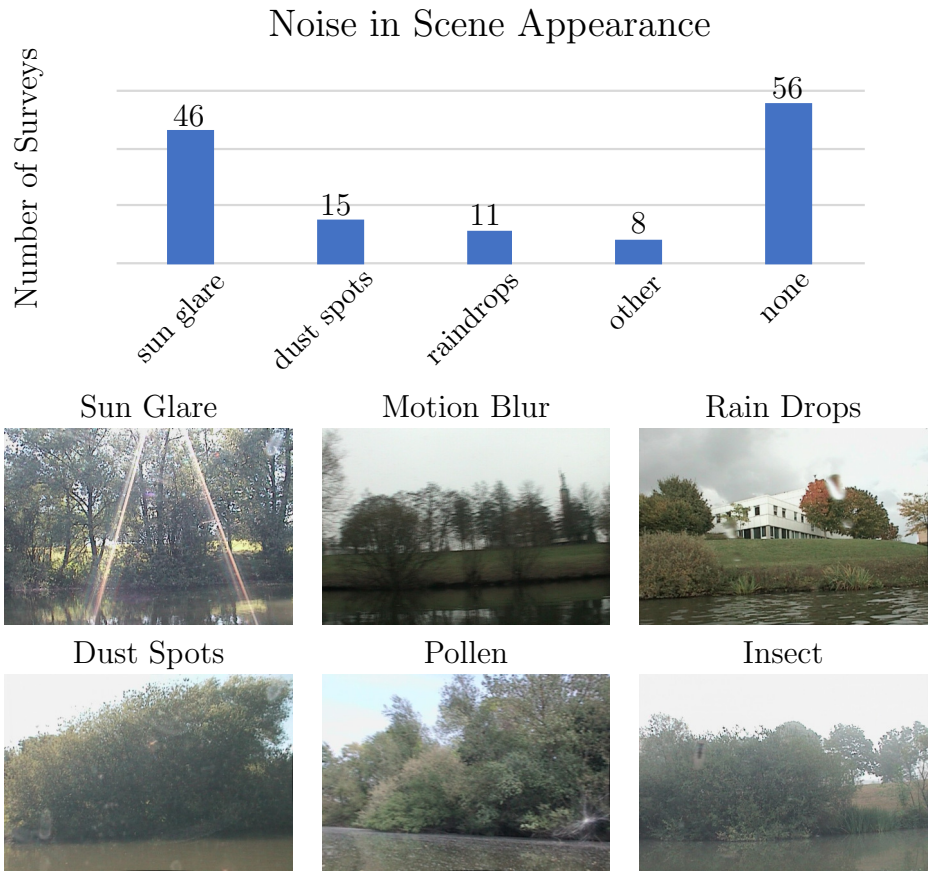

Figure 10: Significant noise was present in many surveys. Sun glare was worse than other types of noise in terms of how much it changed images, how many surveys it was present in, and how many images per survey it affected. Specular reflections on the camera dome are apparent in many images on days of strong illumination. Occasionally, other types of noise obstructed the camera view (raindrops, pollen, insects).

Water does, however, add scene context.

Several kinds of noise also add to the variation in appearance of the images (see Fig. 10). They typically show up as sun glare, distortions, or occlusions. Sun glare was the most prominent (per image and for how long it affected the images). It reduced image contrast and also caused other lens flare artifacts. Dust spots were also often visible in surveys with strong illumination. Debris like pollen and insects sometimes occluded the scene.

3.4 Discussion

Symphony Lake Dataset is novel as a robotic vision dataset because it captures an unstructured environment as it evolved week-to-week over three years (see e.g., [47, 123, 165, 119, 158, 34, 153] for examples of more structured, less frequent, and/or shorter time span datasets). A natural environment is dynamic, which means more can change in a smaller amount of time. The Symphony Lake environment has trees, water, birds, and other flora and fauna of a lakeshore, with some buildings in the background. Sometimes a lot of variation occurred between weeks.

This dataset is interesting for the challenge it brings to perception. Data association across surveys would have to address the variation in appearance of a natural environment. Many different approaches have been proposed to improve condition–invariance in different environments (see e.g. [105] for a review of methods for place recognition). In contrast to indoor environments and suburban streets, the most persistent feature of a natural environment may be its 3D structure (which this dissertation shows in subsequent chapters).

This dataset also presents a challenge to the SLAM community. The size and the number of surveys requires scalable optimization. Each survey potentially has hundreds of thousands of landmarks and thousands of poses. Because standard local image features lack robustness to the variation in appearance, multiple sets of landmarks may have to be used to represent the environment. Optimization must also deal with incorrect correspondences and loop closures (as in [157, 130, 96, 134]).

Success in these spaces could enable work towards identifying and characterizing changes in natural environments. The following chapters used this data as the basis for research on visual data association in a natural environment. The next chapter starts by describing how the structure of Symphony Lake can be captured using visual SLAM.

CHAPTER IV

COMPUTING THE STRUCTURE OF A NATURAL ENVIRONMENT IN ONE SURVEY*

Existing methods for simultaneous localization and mapping (SLAM) are explored in this chapter to find an approach that can extract the environment structure for use in environment monitoring. As a seeing, mobile machine moves through an environment, sensor data may be collected from GPS, a compass, an IMU, a camera, a laser–range finder, etc. Measurements of scene motion extracted from each image along with other sensor values can indicate the camera trajectory and the environment structure. Each measurement may provide a constraint on a particular camera pose or a landmark position. SLAM provides the framework that agglomerates that information into an output trajectory of 6D camera poses and a map of 3D world points.

Approaches to SLAM that are more suited to environment monitoring may be those with better coverage and accuracy in capturing the structure of a natural environment. There are many variants of SLAM, which vary in their ability to provide environment coverage, matching capability across surveys, real–time operation, scale to larger environments, robustness to errors in sensor measurements, etc. It is unclear which set of methods may be more suited to environment monitoring. It is clear, however, that real–time operation is probably unnecessary for surveying. Because the hypothesis of this thesis is that the structure of a natural environment may facilitate robust data association across surveys, an approach is sought which captures good coverage of the environment in an accurate 3D map.

*This chapter is a paper that was presented at the 2014 International Symposium on Experimental Robotics (ISER) [56].

For a map to be representative of a natural environment, a new map may have to be extracted for each new survey. A map should facilitate the primary goal of environment monitoring across multiple surveys captured over a long period of time. A single map would be adequate if it were representative of all the surveys. Yet, that may be unlikely because, unlike a metro or an urban environment of buildings that stay the same over time, a natural environment may experience a large degree of change during the period of observation.

This chapter evaluated two different approaches to visual feature extraction in a framework for pose–graph visual SLAM. Pose–graph visual SLAM was used because it provided a trajectory and a map that was optimized across all the input measurements. Two well–established approaches to visual feature extraction were explored to identify the one better suited to a natural environment. First, the limitations of scene reconstruction using SIFT features was determined before transitioning to an approach based on using Kanade–Lucus–Tomasi (KLT) feature tracking. Although SIFT descriptors provided some matching power between surveys, the matching power was limited, it diminished over time, and the environment coverage was sparse. Scene reconstruction using KLT provided better coverage and longer feature tracks, which resulted in better maps of a natural environment.

4.1 Scene Reconstruction Using SIFT

Local image features, like SIFT (a 128-byte feature of local image gradients), have been a popular choice for use in an environment map, and are explored here for use in mapping a natural environment. When a location is revisited (either in the same survey or in a different survey), the stored feature descriptors from a prior visit could be matched to localize the new pose. When that happens, no new features would have to be extracted to represent the environment. That approach can work if SIFT descriptors provide enough matching power between images of a natural environment.

Figure 11: Snapshots of a scene along the lakeshore. The colored dots are the only features that could be matched. **top left)** The result of feature matching for two nearby images in the same survey. **top right and bottom)** The result of feature matching between this image and the top left image.

For a given image, $\mathcal{I}_a^j$, SIFT features were extracted (using OpenCV [11]) and then matched with the descriptors at image I_{a-1}^j by comparing their feature sets. SIFT descriptors were extracted only at the most discriminative areas in each image, which were typically high–contrast (e.g., buildings). Given matching points between two images, the 3D position was estimated using Hartley and Sturm's iterative linear least–squares triangulation method [63]. Triangulated points provided a quick view of the environment coverage.

4.2 Feature Matching Experiment

A small feature matching experiment was conducted to test how well SIFT feature descriptors could be matched within and between surveys. SIFT feature matching was applied to the scene shown in Fig. 11. Many more matches were found between images from the same survey than from two different surveys. Between surveys, far fewer matches were found between images. Additionally, the building was disproportionately represented, which was expected given that its appearance had much less variation in appearance. Local image feature matching using SIFT is, therefore, much better suited to images of an urban environment than a natural environment.

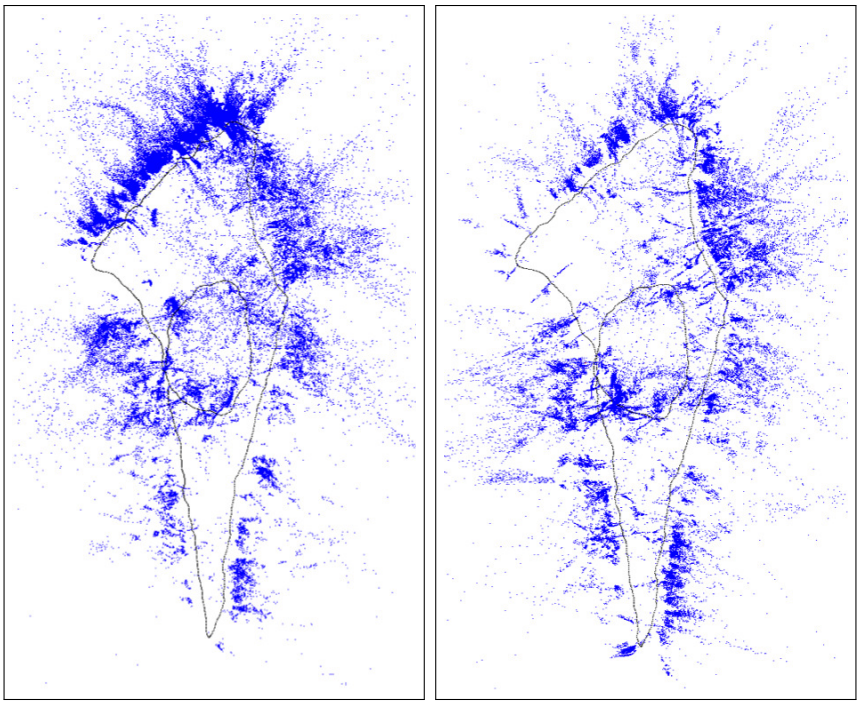

Figure 12: Initial estimates of the point clouds of SIFT features from two different surveys. The Point Cloud Library was used to visualize each point cloud [144].

Environment coverage was next evaluated to determine if SIFT features adequately captured the structure of a natural environment within each survey. The 3D point clouds for two different surveys are shown in Fig. 12. Each map has enough points to show that the global environment structure has been captured. But there are clear gaps in the coverage. Noise is apparent in a good portion of the 3D position estimates due to the fact that optimization has not yet been applied.

The point clouds capture much of the 3D structure of the lakeshore, but due to the sporadic coverage of SIFT feature matching, some locations were better represented than others. Many points were identified on high–contrast areas like buildings, tree–tops, and terrain transitions. The point cloud is thinner in one survey due to the overcast weather. In general, areas with fewer points are either not illuminated well, are part of the featureless grassy bank, are not viewed by the camera, or have high sun glare. These shortcomings suggest SIFT would not be suitable for matching across surveys or capturing the structure of a natural environment.

4.3 Scene Reconstruction Using KLT

In light of the shortcomings of SIFT, a transition was made to Kanade–Lucus–Tomasi (KLT) feature tracking [106], which was found to provide adequate environment coverage. Instead of extracting and attempting to match feature descriptors for each image, the algorithm finds salient keypoints in the first image and then tracks them using optical flow. The Harris corner detector was used to identify salient keypoints. Several constraints were applied to acquire an accurate set of tracked keypoints, which covered the image.

Feature extraction per survey involved identifying up to 300 regularly distributed keypoints, $\mathcal{M}_t^j = \{m_\psi^{j,t}\}_{\psi=1}^{n_{j,t}}$, in an image, $\mathcal{I}_t^j$, and tracking them for the duration they were visible (with an average accuracy of approx. 3 pixels). An image was first subdivided into a 12x20 grid (see Fig. 13) to identify where to extract new keypoints and where existing keypoint tracks were likely to be found. New features were extracted from each image and then sorted into the cells of the grid. Features were only retained if the cells in which they were found had no prior features. Up to five Harris corners in empty grid cells identified new landmarks. The search for matching points in new images was limited to a neighborhood of cells, which ensured efficient computation time.

The end of a feature track was reached in several ways:

- a match was not found in the next consecutive image
- the displacement of the feature was inconsistent with the known camera displacement (i.e., vertical displacement or large overall displacement)
- the feature moved into a grid cell with too many features (more than 5), which typically occurred when background features became occluded
- the feature was marked as an outlier in a RANSAC–based fundamental matrix estimation.

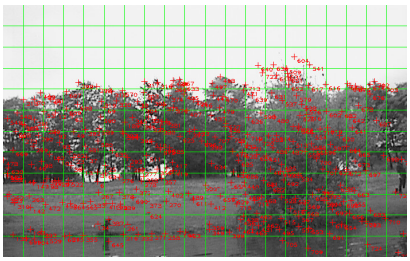

Figure 13: Dense feature set and the grid used to enforce a relatively homogeneous feature distribution. The red numbers identify each feature and its ID.

Results with KLT showed that it was able to cover the full natural environment. The performance of feature tracking in individual images is depicted in Fig. 14. The number and the length of the tracks show that they were reliable and stable over many images. The distribution of features across the images also shows that the scene is well–represented. The 3D landmark coverage across the whole natural environment is shown in Fig. 15. The point cloud clearly indicates that KLT covered the whole environment, particularly both lower and higher contrast areas, and were well–distributed throughout a survey.

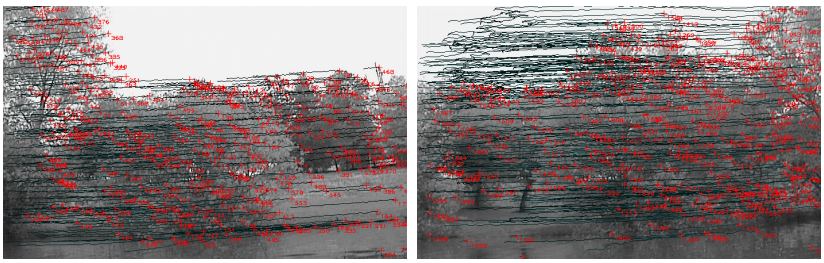

Figure 14: Feature tracks (black) over two different sequences of 50 images. The red text identifies each feature and its ID. The length of each black line indicates the length of the feature track up that point.

4.4 Pose–Graph Visual SLAM

Pose–Graph visual SLAM is the process of acquiring the map and the camera trajectory for each survey. In this formulation, each survey has its own map and camera trajectory, which is independent from those of other surveys. Because KLT provided

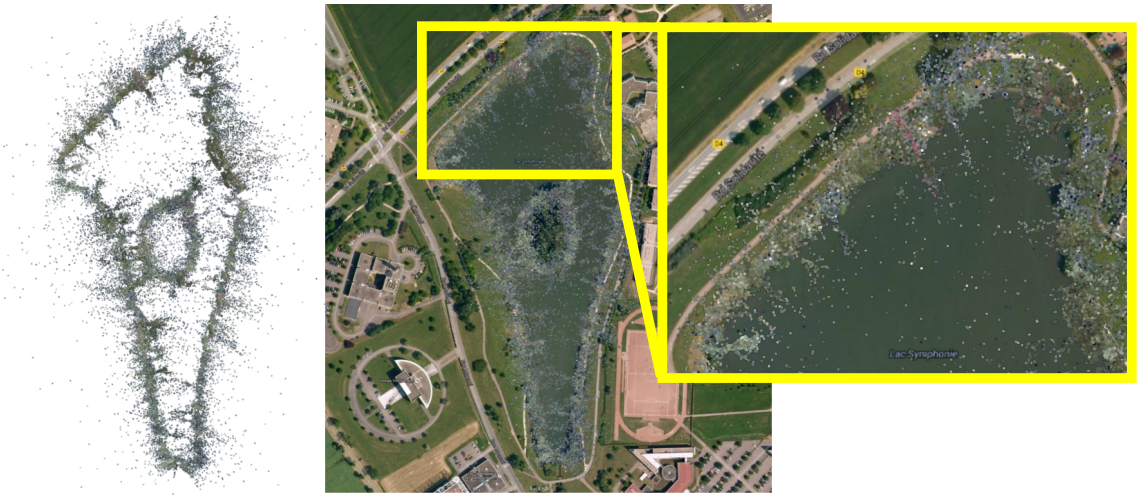

Figure 15: The initial estimate of the map of landmarks that correspond to KLT feature tracks.

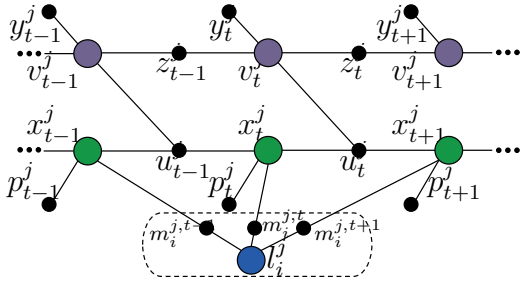

Figure 16: Factor graph of the single–session SLAM optimization problem. A colored node corresponds to a variable to be optimized. A black node corresponds to a factor, which is a constraint on the values of its connected variables. The dotted line depicts a smart factor, which encapsulates a landmark variable and its factors.

time–dependent features, which had little matching power between surveys, no attempt was made here to try to find or apply constraints on the visual observations between surveys. Sensor fusion was limited to the data from one survey of Symphony Lake.

Sensor fusion of the landmark feature tracks, measurements of the camera poses, and prior knowledge of the camera motion is represented using a factor graph of all the measurements and constraints (see Fig. 16). Variable vertices (colored) are the values to be optimized and factor vertices (black) constrain the values of the variables they connect to. The variables include the camera poses, x^j_t, the camera velocities,

v_t^j, and the landmark positions, l_i^j. The factors are derived from measurements of the camera poses, p_t^j, of the change in p_t^j and of the IMU, y_t^j, of the USV's relatively constant speed, z_t^j, and of the landmark feature tracks, $\mathcal{M}_t^j$. Our assumption that the boat moves with constant velocity is used to form a kinematic constraint, u_t^j, which defines the boat's change in pose. Fusing the information in this form enables a fast optimization for a low–error variable assignment.

The optimized estimate of each pose, $\hat{x}_t^j$, velocity, $\hat{v}_t^j$, and landmark, $\hat{l}_i^j$, in the factor graph is found using bundle adjustment. Bundle adjustment simultaneously refines the values of the 6D camera poses, the 6D camera velocities, and the 3D landmark positions to reduce the total error. The nonlinear minimization of error proceeds using the Levenberg–Marquardt algorithm. The GTSAM framework was utilized to perform this step [29]. Within the same framework, smart factors were also utilized, which employ the Schur complement to partition landmarks from poses, and thus yield a more robust result in less time [19].

The result of this procedure for the j^{th} survey is the set, $\Pi_j = \{X^j, V^j, L^j\}$, for $X^j = \{\hat{x}_t^j\}_{t=1}^{n_j}$, $V^j = \{\hat{v}_t^j\}_{t=1}^{n_j}$, and $L^j = \{\hat{l}_i^j\}_{i=1}^{N_j}$. Because measurements of the absolute pose location and orientation (from the GPS and the compass) were captured, each map and the trajectory was consistent with the true scale and orientation of the environment.

4.5 Evaluation

The use of pose–graph visual SLAM with KLT to capture the structure of a natural environment is shown in two different ways. First, the reprojection error of map points was calculated for multiple surveys (Section 4.5.1). The reprojection error is the pixel error of a reprojected 3D landmark with its tracked location. The average reprojection error is small if the optimization is successful. Second, visualizations of the optimized point cloud for one survey are shown (Section 4.5.2). The point cloud

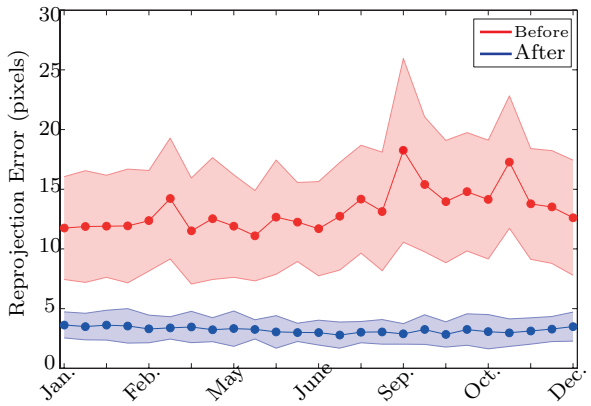

Figure 17: Average reprojection error before and after applying visual SLAM to surveys from 2014.

was overlayed on the satellite view of the environment to show that they matched well. A side–view of the 3D point cloud was generated to show that it corresponds to the 3D environment structures.

4.5.1 Trajectory and Map Accuracy

Feature tracking and optimization were successful was the average reprojection error of landmarks was low (approx. 3 pixels). The reprojection error of a landmark is the L_2 distance between its reprojected 2D position and its original 2D pixel location. The comparison is shown in Fig. 17. The reprojection error consistently averaged 12-15 pixels before optimization and 3 pixels after. The standard deviation was also much higher before optimization.

4.5.2 Point Cloud Visualizations

The function of optimization to make the map consistent with scene structure is apparent in a visualization of the map overlaid with the scene. An example is shown in Fig. 18. Most of the landmarks aligned well with the environment. The alignment is particularly good along the shoreline, where the camera was focused during its trajectory. Some of the landmarks had low reprojection error, but were inaccurate representations of the true 3D structure of the world at the locations where they were

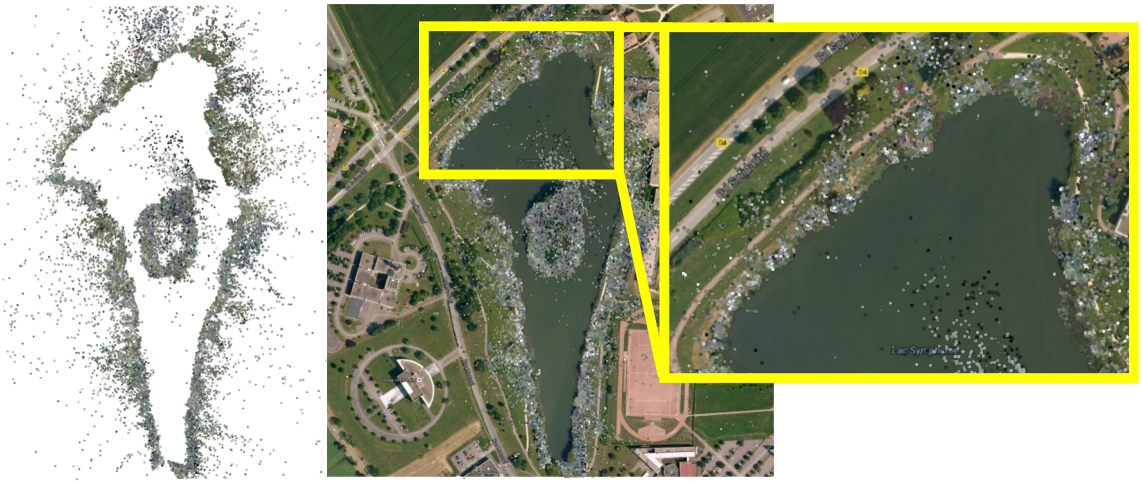

Figure 18: The map of landmarks that correspond to KLT feature tracks after bundle adjustment. The background is the satellite view from Google Maps.

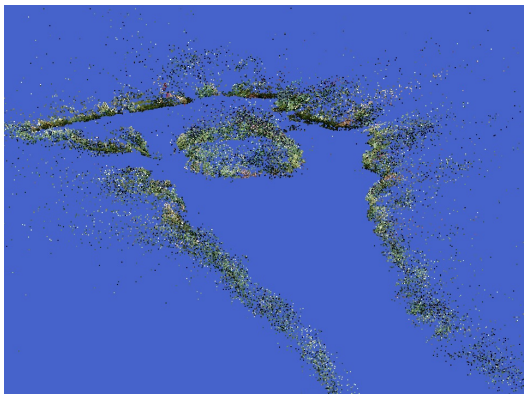

Figure 19: 3D point cloud of Lake Symphony from one session.

tracked. Those cases were more likely for shorter tracks. That occurred, for example, for many points on the floating tree on the north-east side of the island.

A 3D visualization of the point cloud is used to see how well 3D scene structure is captured. An example is shown in Fig. 19. The visualization shows the height of the water is consistent; no part is skewed. Some features are fairly clear, like the fact that there was a floating tree on the north-east side of the island. In most places, however, the sparsity of points was too low to capture the full detail of each scene. Background features were also occluded in many places due to the camera's view at its position 10m from the shoreline.

4.6 Conclusion

Pose–graph visual SLAM with KLT was found to reliably capture the structure from one survey of a natural environment in a trajectory and a map. KLT points were regularly distributed in each scene, which led to full environment coverage after triangulation and optimization. After optimization was applied, point clouds were obtained with low average reprojection error. They also matched environment structures well. These results indicate the structure of a natural environment was successfully captured for each survey.

Although this chapter determined how a map could be computed for each survey, it did not yet connect the observations between surveys. Towards that, the next chapter addresses how images of the same scenes may be identified, using both the visual SLAM result for each survey, and appearance–based methods, which compute the similarity of visual feature descriptors between images.

CHAPTER V

IMAGE RETRIEVAL*

Acquiring aligned images between multiple surveys begins with *image retrieval*, a process to identify images of the same scenes. Image retrieval is a correspondence problem, yet it is a coarse one at the image level rather than the pixel level. The variation in appearance of a natural environment affects both types of correspondence. This chapter seeks to identify a suitable method for image retrieval across a full year of surveys of a natural environment.

Uncertainty in the image retrieval task can stem from the fact that there are multiple ways to identify images from the same scenes. Given that survey images may have the same reference frame (due to GPS and a compass), the corresponding image in another survey would likely be near the one from the closest viewpoint with the most geometric overlap. However, this choice disregards visual information, which may favor a different image. Visual feature descriptors could be used to account for salient features, occluders, different camera angles, and noise (e.g., sun glare), which could help to identify the closest image of the same scene. Yet, relying on visual descriptors for image retrieval could be susceptible to error due to the variation in appearance over time.

This chapter evaluated pose– and appearance–based coarse alignment methods to address image retrieval between surveys of a natural environment. Appearance–based alignment was implemented as video sequence alignment, which consisted of sequence–long matching optimization for a better alignment. Three different visual feature descriptors were compared, including SIFT Flow. With hand–labeled images

*This chapter is part of a paper that was published in the 2017 Journal of Field Robotics (JFR) [59].

for the ground truth, pose–based alignment was found to outperform an appearance–based one for images of a natural environment. Among appearance–based methods, SIFT Flow as an image descriptor was shown to have the most matching power over time. The results collectively indicated that an image retrieval approach based on visual SLAM and SIFT Flow would be suited to images of a natural environment.

5.1 Pose–Based Sequence Alignment

Pose–based coarse alignment identified similar viewpoints between two surveys using the estimated boat pose, as computed for each survey in Chapter 4. The pose–based alignment utilized the fact that surveys were captured in the same global reference frame defined by the GPS and the compass. Two images of the same scene are near those whose poses have the most similar position and heading. Because two different metrics described the position and the heading, however, a single metric of the most similar pose was defined. Two viewpoints captured the same scene if they observed the most similar set of scene points. For this chapter, each set of scene points was defined as a 2D semi-circular grid 10m from each camera. For a given reference pose, the most similar scene was the one whose scene points most overlapped that of the reference pose. Both the raw pose measurements (from GPS and compass) and the optimized estimates (from visual SLAM) were used in the evaluation of pose–based sequence alignment.

5.2 Appearance–Based Sequence Alignment

5.2.1 Video Sequence Alignment

The property of environment surveys that they are approximately captured along the same path made them suitable for video sequence alignment. Video sequence alignment is concerned with establishing correspondence across an entire sequence, rather than image-to-image. The information in a sequence is used to help reduce finer–grained matching ambiguity. As the sequence correspondence is optimized,

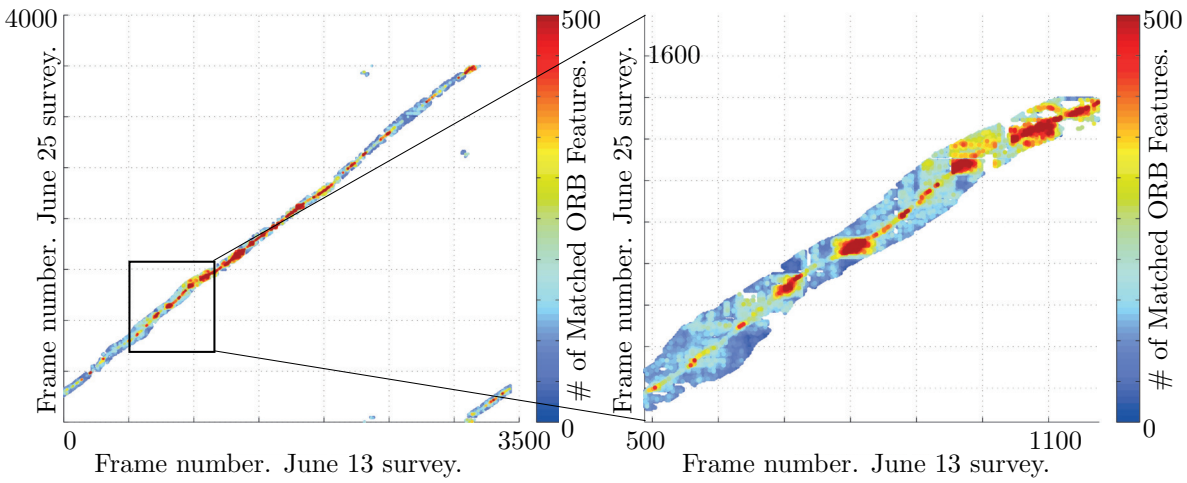

Figure 20: Similarity matrix for the sequence alignment of the June 13th and the June 25th surveys. Only the images at roughly the same locations in both surveys were matched. The descriptor used for this figure was the number of matched ORB features between two frames, colored here from dark blue (least similar) to dark red (most similar). Areas with excessive pose differences are colored white. The close-up on the right shows a that a ridge is identifiable within the area where images were matched, which corresponds to the best matching images between the two surveys. In the cases where the best matches were not always obvious (where the ridge was blue), nearby salient matches (where the ridge was red) helped to guide sequence alignment.

frames with salient features guide the correspondence search across frames with less discriminative features. The best image-to-image matches are those from the sequence whose alignment has the least matching cost (see Fig. 20).

There are multiple approaches to video sequence alignment. The problem has been framed as finding the best path through a difference (or similarity) matrix, which is computed using pairwise matching between each image from two sequences. [168] proposed a method for finding the best path in the similarity matrix based on time–warping. [124] have extended the work of [168] to evaluate different visual feature descriptors on sequences from urban environments, including Histogram of Gaussians (HoG) features and CNN features. Once a distance matrix of costs is computed, the optimal best path between a start and an end node can be solved using the A* search algorithm.

For the evaluation in this chapter, computing the full cost matrix was avoided for

faster computation time. The globally consistent reference frame (due to GPS and the compass) eliminated the need to match all image pairs between surveys. Visual feature descriptors were only matched between images from a similar camera pose (within 20 m and 20 degrees).

5.2.2 Visual Feature Descriptors

Various visual feature descriptors could be used in the appearance–based search for matching scenes. The evaluation in this chapter included descriptors based on ORB, a CNN, and SIFT Flow, as described in Section 5.2.2.1, Section 5.2.2.2, and Section 5.2.2.3, respectively. Including three quite different approaches in the analysis also provided an indicator of which kind of visual feature descriptor may work best for an image retrieval task in a natural environment.

5.2.2.1 ORB Features

Local image features including SIFT, SURF, and ORB are among the most widely used visual feature descriptors, which could potentially work well for image retrieval. One metric for image retrieval is maximizing the number of matched local image features between two frames. The performance of all feature detectors in OpenCV (e.g., SIFT, SURF, ORB, BRIEF) were tested. ORB was used because it performed best. Because the number of matched ORB features is maximized between more similar images, sequence alignment is over a similarity matrix.

5.2.2.2 CNN Features

Feature descriptors from a convolutional neural network (CNN) proved effective across variation in viewpoint and condition [158] in urban environments, which warranted their evaluation for image retrieval in a natural environment. A CNN acquires a generic set of visual feature descriptors across the layers of its network as it is trained for a specific task. Sunderhauf *et al.* [158] showed that the features from a network

trained for image recognition could also work well for image retrieval. An image is provided as input to the network as it would be for recognition, but the feature activations are taken from a specific layer of the network. A particular layer of a network may learn whole–image descriptors, which can be used for image retrieval.

Many pre-trained neural networks exist for different correspondence problems, all with varying abilities of its descriptors to work for image retrieval on images from a natural environment. Following [158] and [124], a preliminary test of features from three pre-trained CNNs in the Caffe library [76] included: 1) the network topology of the BAIR Reference CaffeNet; 2) a network pre-trained on a database of places [173]; and 3) a hybrid network pre-trained on both the place database and a collection of objects. An input image was fed–forward into the network to get the activations at each layer of the network. The normalized dot–product was used to computed the distance between two descriptors from the same layer. An evaluation using the sequence alignment framework showed that features from layer 3 of the hybrid network performed best, which were the ones used for the comparison. This method is referred to as *Conv3* in the evaluation. Because matched conv3 features represented the distance between two images, sequence alignment was over a distance matrix.

5.2.2.3 SIFT Flow (low-res)

An approach to dense correspondence, SIFT Flow from Liu *et al.* [100], was also considered here as a visual feature descriptor for image retrieval. Although dense correspondence is meant to align whole images with pixel–level accuracy, it provided a quite different approach to establishing visual correspondence. SIFT Flow in particular uses an optimization to produce correspondence at the pixel level (this algorithm is explicated in Chapter 6). Additionally, SIFT Flow has been shown to find accurate dense correspondence between some images of a natural environment, which indicates that it may have more matching power than the other two approaches.

SIFT Flow uses a pyramidal matching optimization, which begins at a very low resolution (44×30), which is what is used for image retrieval. Running SIFT Flow at this lowest resolution is fast (less than 1 second) and provides a correspondence score—the minimized alignment energy. One image pair with lower alignment energy than another may align better and/or more closely capture the same scene. Thus, for a given reference image, the aligned image with the lowest alignment energy was taken to be the one that best matched the same scene. Sequence alignment using SIFT Flow was over a distance matrix. More details on SIFT Flow are found in Section 6.2.1.

5.3 Evaluation

Three experiments were used to evaluate image retrieval in a natural environment: Section 5.3.1 evaluates image retrieval between two consecutive surveys; Section 5.3.2 evaluates image retrieval over time from one survey to all the others captured in the same year; and Section 5.3.3 evaluates image retrieval for the best visual feature descriptor over time.

Image retrieval was evaluated across 33 different surveys from the Symphony Lake Dataset, which spanned fourteen months. A survey was part of this set if it was captured between January 2014 and February 2015 and consisted of a fairly complete run around the lakeshore and the island. Surveys were down–sampled to one image every three seconds. The computation of $27 \times 26 = 702$ similarity matrices for each visual feature descriptor took approximately five days on 10 Intel Xeon 8 core PCs.

Image retrieval was evaluated using human–selected images as the baseline. Unfortunately, the Symphony Lake Dataset is too large to hand–align all the surveys. Therefore, two reference sets were created: 1) for the complete survey alignment of the June 13 and the June 25 surveys (Sec. 5.3.1); and 2) for the partial survey alignment of the June 25 survey to the other 27 (Sec. 5.3.2). The second set consisted

of two separate sections of the lakeshore (the first with more vegetation, the second with a building in the background). Each consisted of 200 labels (for a total of $27 \times 200 = 5400$ labels) whereas the complete survey consisted of about 4000 labels. To speed up the hand–labeling task, a user started from the closest image as determined by the GPS and the compass, and then scrolled forward or backward through the survey until they found the image pair that matched best. After establishing that visual SLAM consistently provided an accurate image retrieval, that was used as a baseline for the last evaluation in Section 5.3.3.

Note that the absolute alignment error of image retrieval in Figures 21–24 is measured in seconds. Lower is better. Error using a time metric indicates how synchronized the two video sequences are given that they are aligned using a dynamic time warping sequence alignment. The value is the offset from the ground truth, where the ground truth corresponds to some specific frame and the offset is some number of frames in the 1 Hz video sequence away from that. An error of, for example, five seconds corresponds to a difference of five frames between the selected frame and the ground truth frame.

5.3.1 Image Retrieval Across Two Surveys

The accuracy of image retrieval between two consecutive surveys is shown in Fig. 21. The methods all performed about the same on average, with the exception of the Conv3 descriptor, which performed much worse. SIFT Flow showed a very slightly better accuracy than the others. Yet, the approaches all deviated from the ground truth fairly consistently over the complete survey.

Deviation among the methods occurred at a couple scenes, however, which was likely due to the subjectivity of the hand–labeled ground truth. There was deviation at about image 1380, for example. Outliers occurred there because the operator took control of the boat to avoid fishing lines. During the maneuver, the boat was

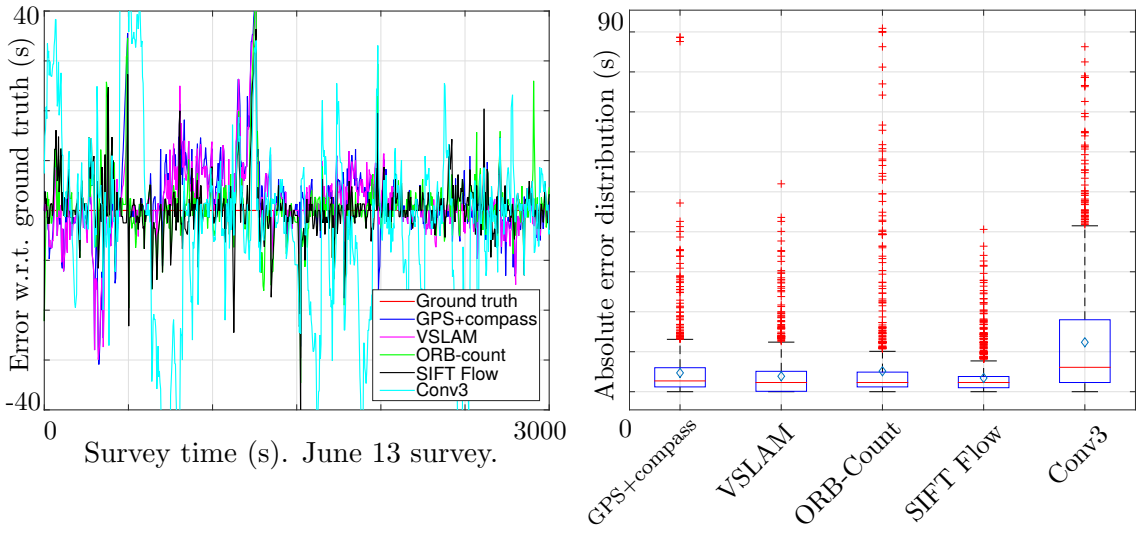

Figure 21: Alignment accuracy of five different methods on the sequence alignment of surveys from June 13 and June 25, using manually labeled images as the ground truth. Accuracy is measured in seconds offset from the ground truth, which corresponds to the number of frames in the 1 Hz video sequence. **left)** Accuracy per method over time. **right)** Distribution of the absolute error over the full survey for each method. The red line indicates the median of the absolute error and the diamond the mean.

moved somewhat perpendicularly to the shore, which is inconsistent with how the boat moved on its own in the other surveys. The hand–labeled image there was more subjective.

The little variation in appearance that accumulated between the two surveys in two weeks helped the appearance–based approaches retain descriptive power. Some areas were, however, easier to match than others due to the availability of more discriminative features, like buildings (See Fig. 20). The Conv3 features lacked descriptive power for images of vegetation. This is consistent with the observations of related work (e.g. [124]). As Naseer *et al.* [124] showed, a CNN that is trained for scene recognition in an urban environment would likely decrease the saliency of vegetation in order to boost the recognition of streets and buildings in the midst of seasonal variations. The next section evaluated how these results held up over time.

5.3.2 Image Retrieval Over Time

Although two appearance–based solutions worked well in the case of a short time interval between two surveys, this trend did not hold over time, as shown in Fig. 22 and Fig. 23. Figure 22 is the evaluation for the sequence of shore with with mostly vegetation, Fig. 23 the evaluation for the sequence with a building in the background. Of all five methods for sequence alignment, only the pose–based methods retained a consistently high accuracy over time. Appearance–based approaches sometimes were more accurate (as in Fig. 21), but only slightly and only between surveys captured within approximately the same season. All of the methods had at least some noise compared to the ground truth human labels.

Of the two pose–based sequence alignment metrics, visual SLAM and GPS+Compass performed about the same because they used the same reference frame (the one defined by the GPS and the compass). Visual slam perhaps performed marginally better. Visual SLAM corrected instances of kinematically inconsistent poses from GPS+compass, which would have led to more accurate locations in the shared global reference frame. The lack of loop closures between surveys limited, however, visual SLAM from performing much better than this.

Of the three visual feature descriptors used for appearance–based sequence alignment, SIFT Flow appeared most suited to image retrieval in a natural environment. Yet, each method had large amounts of noise in several places. The accuracy of image retrieval with SIFT Flow and ORB-Count dropped in the same way it did in the results of Fig. 32. The variation in appearance had the same effect on the visual feature descriptors whether the alignment was coarse or precise. Descriptive power was also undermined due to occlusions of the July surveys and sun glare of the August surveys. The conv3 features had little descriptive power in any case, but as already pointed out, that was likely due to the fact that better features could come from a CNN specifically trained for this task.

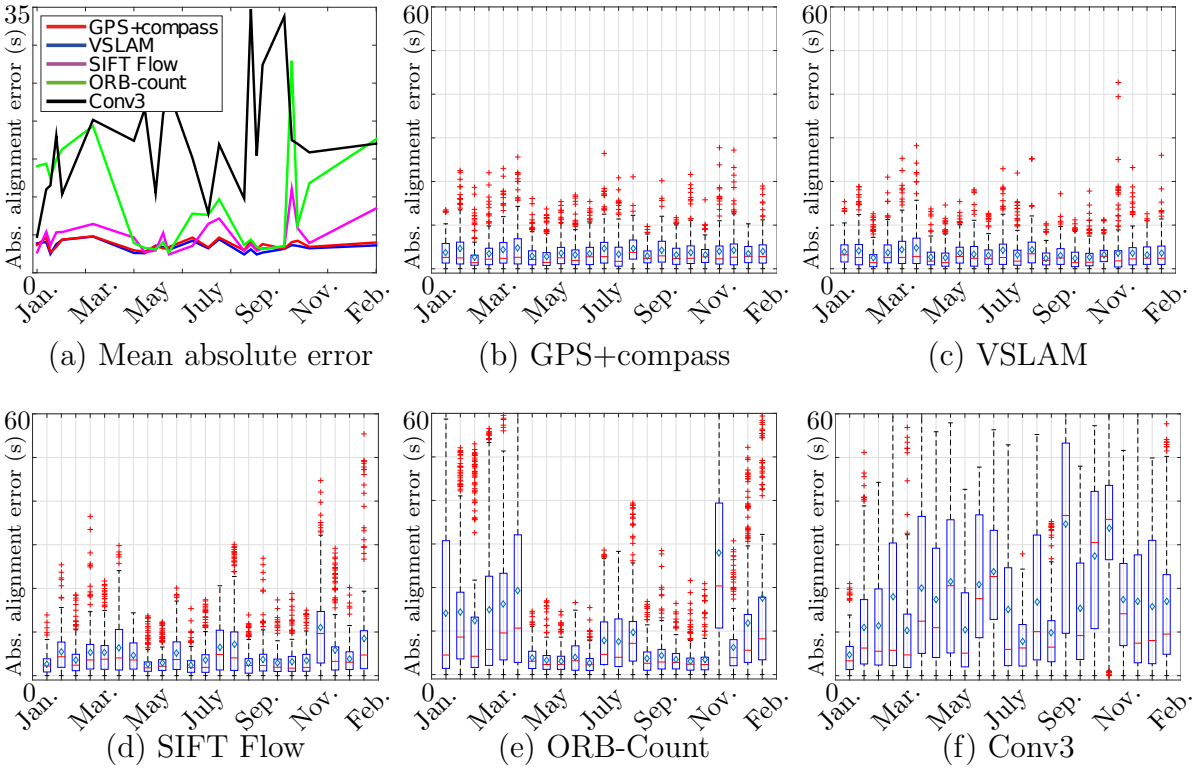

Figure 22: Quantitative comparison of the five different coarse–alignment methods for sequence one, on which the June 25th survey was aligned with 24 other surveys. The accuracy of each method was measured as the offset to the human–labeled matches.

5.3.3 The Best Visual Feature Descriptor For Image Retrieval Over Time

SIFT Flow appeared to outperform ORB-count, but the result was somewhat inconclusive due to the fact that only two sections of shore were evaluated. Both performed similarly in one case, and both had large amounts of noise in many places, which thus warranted a more comprehensive evaluation than those shown in Fig 21-23. Yet, the hand–labeled ground truth data was only available for those two subsets of the environment. Because getting the hand–labeled ground truth for all 702 survey comparisons was infeasible, the visual SLAM result was used for the ground truth. ORB-count and SIFT Flow were compared to that for all survey comparisons across the entire environment. The accuracy of both approaches was calculated as a function of the number of weeks between surveys to capture how robust each method was to

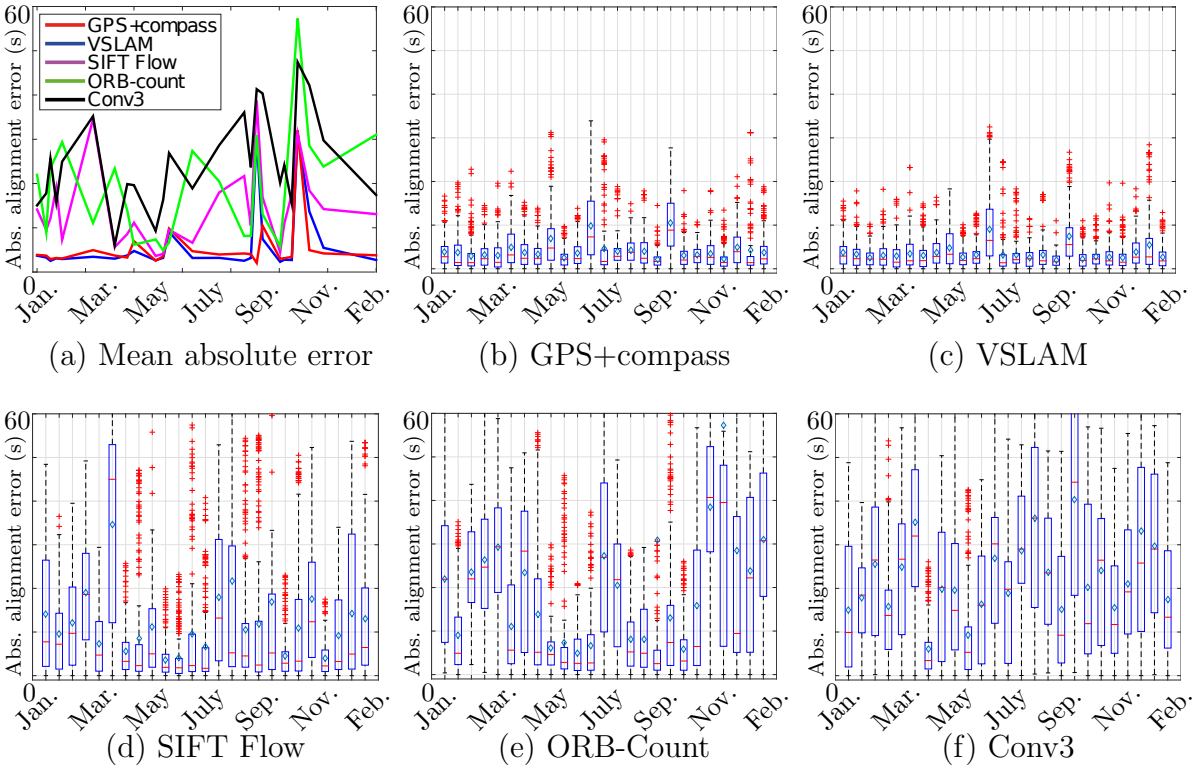

Figure 23: Quantitative comparison of the five different coarse–alignment methods for sequence two, on which the June 25th survey was aligned with 25 other surveys. The human–labeled images provided the ground truth against which the accuracy of each method was measured.

the variation in appearance.

Figure 24 shows the results. The SIFT Flow coarse alignment approach was, on average, more robust than ORB–count to the variation in appearance that accumulated between surveys over time. Whole–image correspondence using SIFT Flow aligned the scene manifold, which apparently retained more descriptive power over time than the aggregation of matched local image features. Both approaches lost, however, matching power over time up to six months later.

The parabolic shape of the curves indicates that images from the same season may align better than images from different seasons, even if the images from the same season were captured from different years. Only a small amount of data supported that conclusion, however, due to the fact that fewer alignments spanned the full year in this comparison. Much of the supporting data came from images from the

alignment of the March 2014 survey and the Feb 2015 survey. The drop in data toward 52 weeks also contributed to the variation in performance near the end of the graph.

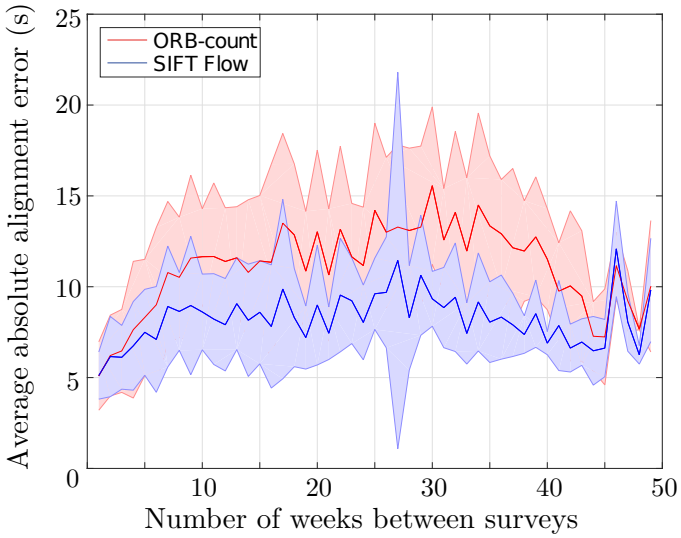

Figure 24: Accuracy of appearance–based coarse survey alignment as the number of weeks between surveys was increased. Visual SLAM provided the reference against which alignment error was measured.

5.4 Conclusion

Given image sequences of a natural environment, with each image associated with a pose prior from sensor data, visual SLAM provided the best basis for image retrieval, and visual feature descriptors appeared only able to improve upon it within a limited range of variation in appearance. Image retrieval using the pose priors from visual SLAM was both accurate and stable across all the surveys in a year. The best of three visual feature descriptors provided, in contrast, comparable or better performance only between surveys within approximately three months of one another. Any improvement over visual SLAM was only likely within that range.

Dense correspondence may be one of the most appropriate methods for data association between images of a natural environment. The manifold structure of the scene captured in a single image was a more stable feature for visual data association. As

local image feature aggregates and activations from a layer of a CNN also both relied on the visual appearance, but performed poorly, the visual feature descriptors for those two methods was less reliable for image retrieval. The higher spatial resolution of SIFT Flow was, apparently, essential to better performance.

Given an approach to automatically find images of the same scenes (apply visual SLAM first and then apply SIFT Flow to refine the coarse alignment in case the images have a similar appearance), the next chapter evaluated the time scales across which SIFT Flow improved upon visual SLAM to provide precise dense correspondence between images of a natural environment.

CHAPTER VI

DENSE CORRESPONDENCE FOR NATURAL ENVIRONMENT MONITORING*

This chapter describes the use of dense correspondence to assist a human in the recurrent inspection of a natural environment. Seeing, mobile machines promise a flood of high resolution data in surveys over large spatial and temporal scales, as in monitoring tasks on lakes, farms, and secured sites. Manual environment inspection may be slow if the task also requires performing manual data association. Any manual labeling effort may be cumbersome, let alone manually searching an entire image for the dense correspondence to another image from a different survey. Yet, it is clear that images captured from a mobile camera would likely add variation in viewpoint. It may be, however, possible to apply methods for dense correspondence to remove the variation in viewpoint and facilitate rapid manual inspection.

Existing approaches for computing the dense correspondence between two images may be able to align images of a natural environment to facilitate monitoring. For an image pair of the same scene, a dense correspondence is the full–resolution, pixel–wise data association from one image to a reference image. A new image can be generated from the dense correspondence, which is one whose variation in viewpoint to the reference image has been removed. The image quality of the generated image depends, however, on the accuracy of the correspondence. Because images of a natural environment are what are to be aligned, which may have significant variation in appearance, it is unclear whether the dense correspondence may be accurate enough

*This chapter is part of a paper that was published in the 2017 Journal of Field Robotics (JFR) [59]. This chapter describes methodology that was first presented at the RSS Workshop on MVIGRO 2015 [54] and FSR 2015 [57].

to facilitate the comparison.

This chapter describes the use of dense correspondence to facilitate rapid manual inspection of a images from a natural environment. Manual inspection was performed both before and after dense correspondence was computed between an image pair. A first experiment was undertaken to show that when an image pair is unaligned, a human may spend a significant amount of time searching for a single corresponding point between two images. After identifying three different methods for dense correspondence, and then the one most suitable for environment monitoring, a large scale study was undertaken to analyze environment monitoring using aligned images. Images of the same scenes were aligned 1) across a year of surveys; and 2) between 10 consecutive surveys. The number of aligned images showed the extent to which dense correspondence may be accurate. The changes identified during the labeling process showed that it may facilitate comparison when it is.

A high degree of variation in appearance must be overcome in the search for corresponding points between images of a natural environment (see Fig. 25). To characterize the difficulty of this search, humans were timed on a manual data association task. Humans searched for a single matching feature between images while they were timed. Image pairs were from different surveys and typically captured the same scenes. A set of 24 surveys were used for this experiment, which spanned the year 2014 of the Symphony Lake Dataset. Given a random image pair from this set, subjects either found and clicked a matching feature in both images or marked the alignment a failure. The latter label was also used if the subject could not recognize a co-occurring feature in both images. For this analysis, 937 valid data points were collected, which is 950 data points minus 13 cases of experiment interruptions (any with a duration longer than three minutes).

Figure 25: Variation in appearance of a section of the lakeshore in the span of nearly a year, captured in six images. There is significant variation in the vegetation, the lighting, the sun glare, and the water level. This makes data association difficult.

6.1 The Difficulty of Inspection Between Unaligned Images Of a Natural Environment

Figure 26 shows the results. Subjects marked 902 (98%) of the image pairs as successfully aligned. Most image pairs provided enough context for the comparison—99% were shifted by less than half the image width and 90% by less than one fourth. Yet, 111 of them (12%) required more than 30 seconds to recognize and then select a single matching feature in both frames. Label time had little correlation with the percent of image overlap, and surprisingly, little correlation with the amount of elapsed time between surveys.

The difficulty of manual image alignment was explainable by the lack of persistent structural features at particular locations along the shore. Examples of hard-to-match image pairs are shown in Fig. 27. Lone trees and buildings were easiest to match; dense foliage was hardest. Although the foliage dropped away in some months, perceptual aliasing across the remaining features made those difficult to match.

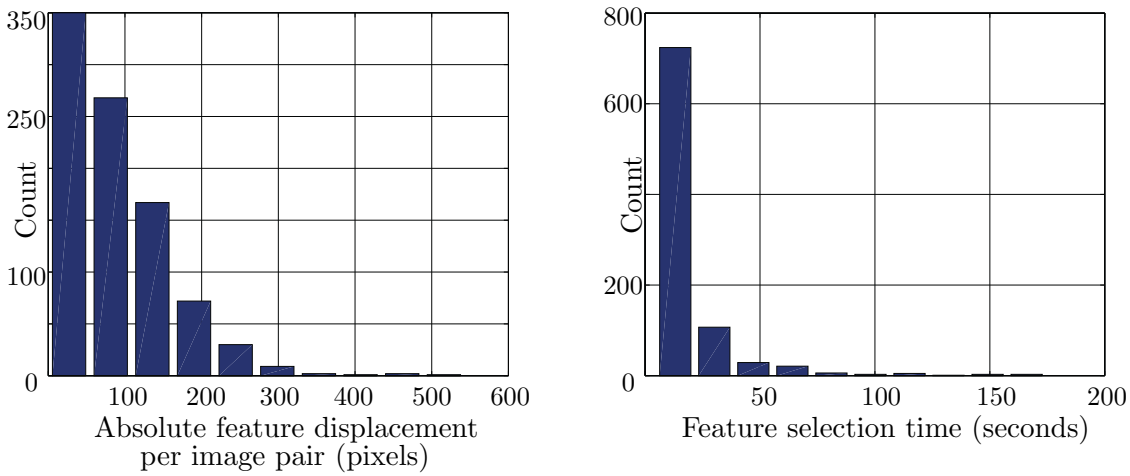

Figure 26: left) The distribution of feature displacements on user–marked image pairs, which indicates most image pairs had significant overlap. **right)** The distribution of time spent labeling the images. Humans took longer than 30 seconds to find a single matching feature in 12% of the image pairs.

Note that subjects almost always found one matching feature point among hard-to-match cases. That suggests at least one matching detail was clear between images. However, precise matches were not always matched. For example, in some cases matches were placed on the center of a bush, which was clearly the same, but whose foliage obscured a precise match. The likelihood of perceptual aliasing and the fact that time–consuming spatial searches were required to find one matching point suggests that point–based feature matching may be inapplicable to natural environments. Instead, correspondence across an entire scene may be better suited to environment inspection.

6.2 Algorithms for Dense Correspondence

An algorithm for dense correspondence takes as input two images, $\mathcal{I}_a^j$ and $\mathcal{I}_b^k$ of a scene, which correspond to the image from survey j at time a and from survey k at time b, and outputs their correspondence, w, at every pixel, which can be used to align them. Figure 28 depicts the computation for one algorithm. Three methods were part of this comparison: Section 6.2.1 introduces SIFT Flow; Section 6.2.2 Deformable Spatial Pyramids; and Section 6.2.3 Daisy Filter Flow. SIFT Flow is

May 2 and May 28 Feb. 5 and June 6 June 6 and Apr. 9

Figure 27: Three image pairs (one per column) that were hard for a human to align. A human spent over 30 seconds on each image pair, both validating that the images capture the same place and then selecting the same physical feature in them.

described in more depth than the other two methods because it was used throughout this dissertation.

6.2.1 SIFT Flow

SIFT Flow [100] combines the accuracy of point–based feature matching with the robustness of whole–image matching by aligning whole images worth of SIFT features. A so–called 'SIFT image', S_a^j, has the same height and width as the original image, $\mathcal{I}_b^k$, but with 128 channels. Each pixel of $\mathcal{I}_b^k$ essentially becomes a 128-byte SIFT descriptor [103]. Local gradient information from the 16x16 pixel neighborhood of the pixel is captured in the feature. Two particular SIFT descriptors match if their L_1 distance is relatively low.

SIFT Flow consists in aligning two SIFT images, S_b^k and S_a^j, which correspond to $\mathcal{I}_a^j$ and $\mathcal{I}_b^k$. The idea is that, although some areas of an image are uninformative, a matched scene manifold—as defined by SIFT features—could anchor an alignment. Two images with very different appearance would be aligned along the manifold, with the uninformative regions taking values in the neighborhood it defined. Thus, a SIFT

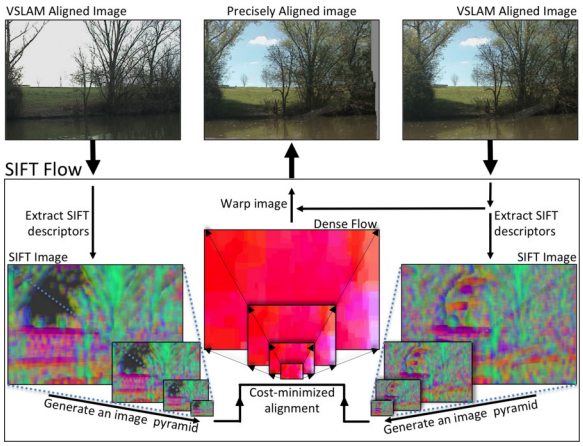

Figure 28: Depiction of SIFT Flow for computing the dense correspondence of two images of the same scene and then using the result to warp one image into precise alignment with the other.

feature is extracted for every pixel of $\mathcal{I}_a^j$, which produces the SIFT image, S_b^k. Two SIFT images, S_b^k, S_a^j, are what are to be aligned.

Image alignment is defined as an optimization using a Markov Random Field (MRF). Each variable in the MRF corresponds to a pixel of S_b^k. Edges connect the variables for adjacent pixels. A pixel, $q \in S_b^k$, is assigned a flow $\mathrm{w}(q) = \{u_q, v_q\}$, where $u_q, v_q \in [-\mathrm{h}..\mathrm{h}]$, and $q + \mathrm{w}(q) \in S_a^j$. The quality of a flow is measured in terms of the descriptor match quality (data), how similar it is to the flow of adjacent pixels (smoothness), and how large it is (regularization), as defined by the alignment energy:

$$
\begin{aligned}
E(\mathrm{w}) = & \sum_q min(|S_b^k(q) - S_a^j(q + \mathrm{w}(q))|_1, \mathrm{t}) \\
& + \sum_{r \text{ adj. } to \ q} min(\alpha|u_q - u_r|, \mathrm{d}) + min(\alpha|v_q - v_r|, \mathrm{d}) \\
& + \sum_q \nu|u_q + v_q|
\end{aligned}
\tag{1}
$$

The minimized alignment energy, $E(\overset{*}{\mathrm{w}})$ is computed using a coarse–to–fine alignment down an image pyramid with four layers. The initial flow field, w, is of images that are downsampled by a factor of 2^4. Whereas the flow field doubles in size with

successive layers, the hypothesis space for each variable shrinks, which telescopes the correspondence. The truncation term, t, has value equal to the median of the descriptor distances between S_b^k and S_a^j. The other parameter values (α, ν, d, and h) are listed in Appendix A.

Note that SIFT Flow has multiple limitations to align scenes due to its formulation. SIFT Flow is not robust to variation in scale or rotation. A SIFT descriptor is typically computed at the image scale where the Difference of Gaussian image has a peak, but for SIFT Flow they are all computed at one scale. The lack of SIFT keypoint detection also results in nonoriented descriptors. The following two approaches addressed these issues with different formulations, which in some cases make those algorithms better.

6.2.2 Deformable Spatial Pyramids

Deformable Spatial Pyramids (DSP) [83] defines a formulation of dense correspondence like that of SIFT Flow, but for faster speed and scale invariance. Rather than optimize for dense correspondence through a multigrid image pyramid, the optimization is defined over a pyramidal graph of variables and edges. A root variable is defined for the whole image and is connected by an edge to four child variables, which correspond to the subdivision of the image by four. The four–way subdivision recurs down to individual pixels. Edges connect parents to children and adjacent variables at the same level. Only the variables that represent the flow at the bottom–most layer do not have edges to neighbors. A scale variable is introduced for each node, which allows each node to be matched at a different image scale.

Optimization for dense correspondence over the DSP graph is much faster than SIFT Flow and is invariant to scale. The improvement in computation time comes from the fact that many fewer variables and constraints are part of the optimization search. A significant boost is also due to the lack of smoothness at the lowest layer of

the image pyramid, where each flow corresponds to the data association of one pixel. The scale term facilitates matching across scale, rather than at a fixed scale. The descriptors are, however, still computed at one fixed scale.

6.2.3 Daisy Filter Flow

Daisy Filter Flow (DFF) [170] adds power to dense correspondence with scale and rotation invariance due to its use of daisy descriptors. A challenge of extracting a descriptor at each pixel is identifying the scale and the orientation for the descriptor. The scale and the orientation are scene–dependent, and in dense correspondence are typically left fixed. DFF gains matching power by enumerating different scales and rotations for each descriptor in the extended search for fine–grain matches. The use of the daisy descriptor and precomputed orientation maps help make the extended search more tractable.

6.3 Experiments

The experiments characterize how well the dense correspondence of images of a natural environment can be computed using existing methods. First SIFT Flow is identified as the best method of three for dense correspondence (Section 6.3.1) and then a small extension is made to try and incorporate scene structure (Section 6.3.2). After a metric is defined for the image alignment quality (Section 6.3.3), images were aligned between surveys increasingly separated in time over a year to find where it fails. Dense correspondence was also used to align images between consecutive surveys to gauge the use of dense correspondence for that use (Section 6.3.5). Section 6.3.6 points out several environment changes that were found due to dense correspondence; Section 6.3.7 examples where SIFT Flow was robust to variation in appearance; and Section 6.3.8 examples where SIFT Flow was limited in its ability to align particular scenes.

SIFT Flow Deformable Spatial Pyramids Daisy Filter Flow

Figure 29: Image registration using SIFT Flow, Deformable Spatial Pyramids, and Daisy Filter Flow. The pixel-level alignment quality was higher using SIFT Flow.

6.3.1 Qualitative Comparison of Dense Correspondence Methods

The three methods for dense correspondence were evaluated by aligning an image pair of one scene between two surveys. The evaluation indicated the image alignment quality and the runtime of each method. The results are shown in Fig. 29. All three methods produced well–aligned images, but the alignment quality from SIFT Flow was highest. Fine details were lost in the DSP result. Several more artifacts were added to the DFF result. DSP finished in under a second, SIFT Flow in approx. 30 seconds, and DFF in approx. two hours.

Of the three methods for dense correspondence, SIFT Flow was chosen for further evaluation because the other two did not appear to produce a more accurate alignment and some of the problems they were designed to fix may be infrequent in monitoring applications. The runtime advantage of DSP over SIFT Flow is partly because it drops spatial coherence in the fine, full resolution level of image alignment. Over half of the full runtime of SIFT Flow is spent aligning the bottom–most layer, which searches a tiny 3x3 hypothesis space around each pixel. That granularity is, however, unnecessary until a full resolution dense correspondence is desired. Moreover, although DSP is robust to variation in scale, during surveying a camera may consistently be the same distance away from the scene (our boat was consistently 10

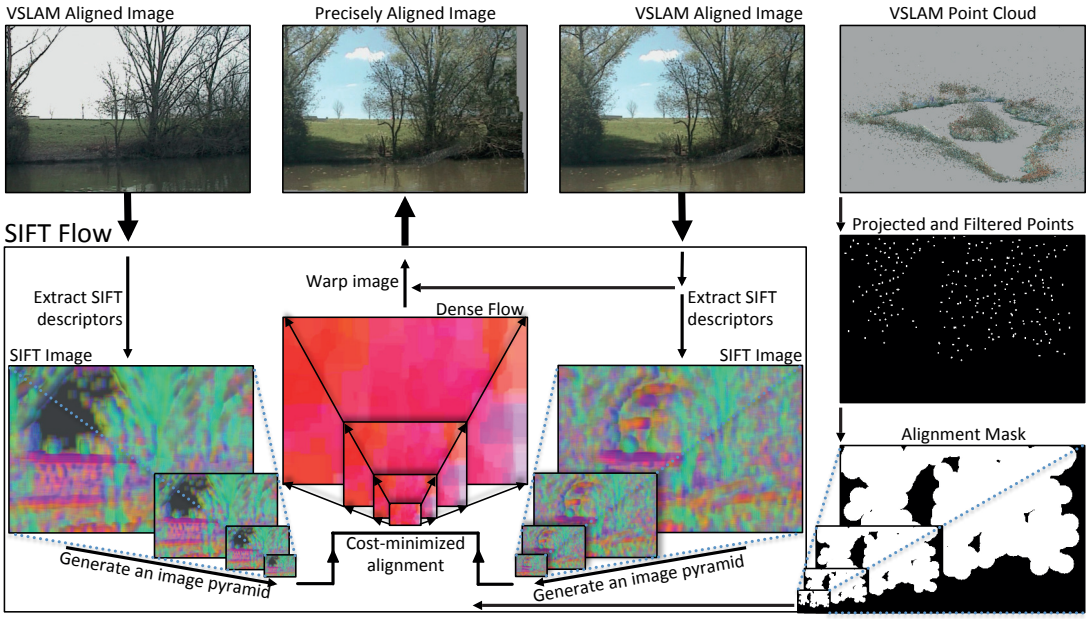

Figure 30: The registration of two images, which visual SLAM found to capture the same scene. For each image, SIFT descriptors are computed at each pixel to form a SIFT image, which is down–sampled into an image pyramid. An image mask representing the lakeshore (derived from the 3D information in the feature tracks of visual SLAM) is used to bias where the SIFT images are aligned, which helps avoid aligning noise due to the sky or the water. The output flow aligns one of the input images to the other, which enables quick change detection for manual inspection tasks.

m from the shore). Similarly, although DFF is robust to variation in rotation, those may be relatively infrequent if an autonomous vehicle is repeating a survey along a similar trajectory. The runtime of DFF also took that out of consideration for future use.

6.3.2 An Initial Use of Scene Structure With SIFT Flow

A mask was added to SIFT Flow to bias the alignment of the lakeshore over the sky or the water, which was an initial exploration into the use of scene structure to improve dense correspondence accuracy, and is the method that was used for the rest of the evaluations in this chapter. As is, SIFT Flow's cost function is designed to align the contents of a scene indiscriminately. Unfortunately, images with a majority of sky and water can obscure an alignment of the rest of the scene because those

areas may retain little consistent structure between surveys. Reflective water can, for example, reduce the likelihood of a good alignment because the reflectivity of the water may change between surveys. The varied appearance of the sky can also affect the alignment.

The bias to SIFT Flow's cost function was derived from the optimized map from visual SLAM (from Chapter 4). The location of the sky and the water in each image was estimated using the landmark positions. Although most landmarks were on the shore because most corner features occurred there, some were occasionally identified in the sky and the water. Points with a negative elevation were typically due to reflections in the water. Points far away were typically in the sky. The rest were interpreted as part of the scene to be aligned. Given an image and the set of keypoints tracked in it, an image mask was created drawing a circle for each non-sky, non-water keypoint (with radius $r = 28$, which gave the best performance). For each pixel in the non-masked region, the data terms of SIFT Flow's objective function were biased (by a factor of 1.5) to align the contents there compared to the other areas of the image.

6.3.3 Alignment Quality Metric

Because the ground truth image alignments were unavailable, a good metric for the alignment quality was found in manual labeling, which involved flickering an image and the one aligned to it back-and-forth. Displaying the aligned image pair in this way enabled rapid manual inspection by drawing attention to changes. Although only a subset of all the aligned image pairs may be selected for inspection, the time spent inspecting every scene in a side-by-side comparison can add up for a large environment. Fortunately, humans are highly sensitive to changes in images of a scene if the images are flickered back-and-forth [131]. Therefore, aligned images were flickered back-and-forth to gauge their quality. In case two images aligned poorly, the

user could have toggled the side-by-side display of the non-warped images.

The quality of each alignment was manually identified according to three criteria:

precise almost the entire image aligned well with little noise

coarse the images correspond to the same scene and some objects may be precisely aligned

misaligned the images correspond to different scenes or it is hard to tell they come from the same scene.

These criteria are shown in Fig. 31.

6.3.4 Alignment Quality Over Time

The primary value of dense correspondence to a visual surveying application in a natural environment may be in how well images are aligned across the variation in appearance over time, which was evaluated here. In this analysis, images from a survey captured on June 25, 2014 were aligned with images from 24 other surveys from 2014. Given image pairs from the same scenes from each survey, SIFT Flow was applied to align them, and their alignment quality was manually labeled.

Note that, to find images of the same scenes, first the cover set of the June 25, 2014 survey was computed (as described in [59]; an example is given in Fig. 31) and then a local search was used to identify the best image candidates for full resolution dense correspondence. Given a pose from one scene of the reference survey and the nearest pose from the aligned survey, a low–res local search was performed around the two candidate poses to find the image pair that aligned best. The search improved the chance that the image pair would align well because slight perspective differences between the two images in many cases led to misalignments. Thus, images at 0, $\pm$ 1.5 , and $\pm$ 3.0 second offsets from the two image candidates were considered, for a total of $5x5 = 25$ different alignments. The image pair with the lowest alignment energy was taken to be the one that aligned best. Full–resolution image alignment was run on this

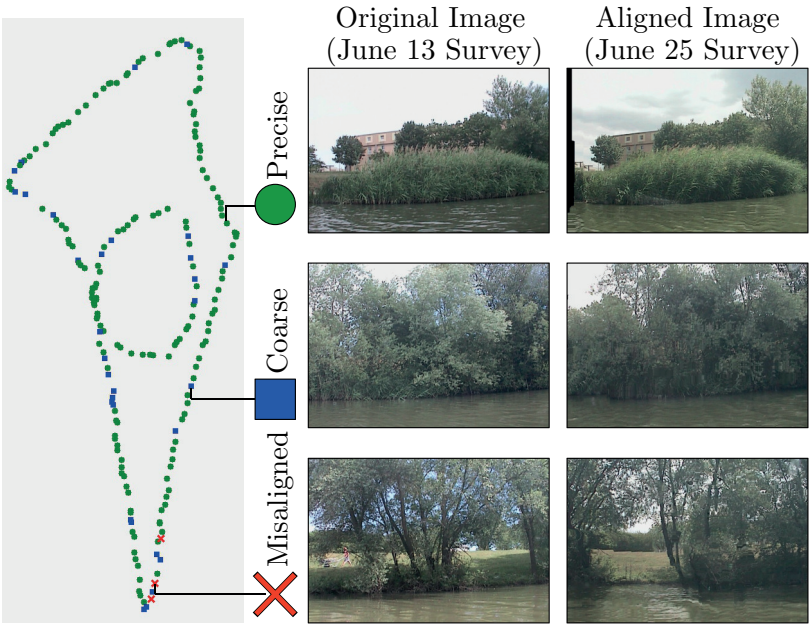

Figure 31: Examples of precise, coarse, and misaligned image pairs from the comparison of the June 13 and the June 25 surveys. The comparison involved computing the dense correspondence of image pairs for the scenes shown (the cover set). Note that the data shown here comes from the evaluation in Fig. 33. The data from Fig. 32 was part of a separate evaluation, which led to slightly different numbers of each alignment quality. Because the misalignment shown is an ambiguous case in which half of each image are overlapping, a human could have alternatively labeled the image pair as coarsely aligned.

image pair. During the search, the dense correspondence was only calculated at the low-res, top layer of the image alignment pyramid, where the alignment energy still indicated the best matching image pair, yet which was much faster (≤ 1 s runtime) than running full–resolution image alignment on each pair (approx. $5 - 20$ s runtime per pair).

The results are shown in Fig. 32. The graph shows that the number of precisely aligned images trends down as more time has elapsed between surveys. The winter surveys had few precise alignments with the mid-summer survey. Although the coarsely aligned images typically captured the same scene, image registration could not align the images well. Large artifacts added a large source of error. Patches of the images often aligned well while other areas aligned with replicated patches. In

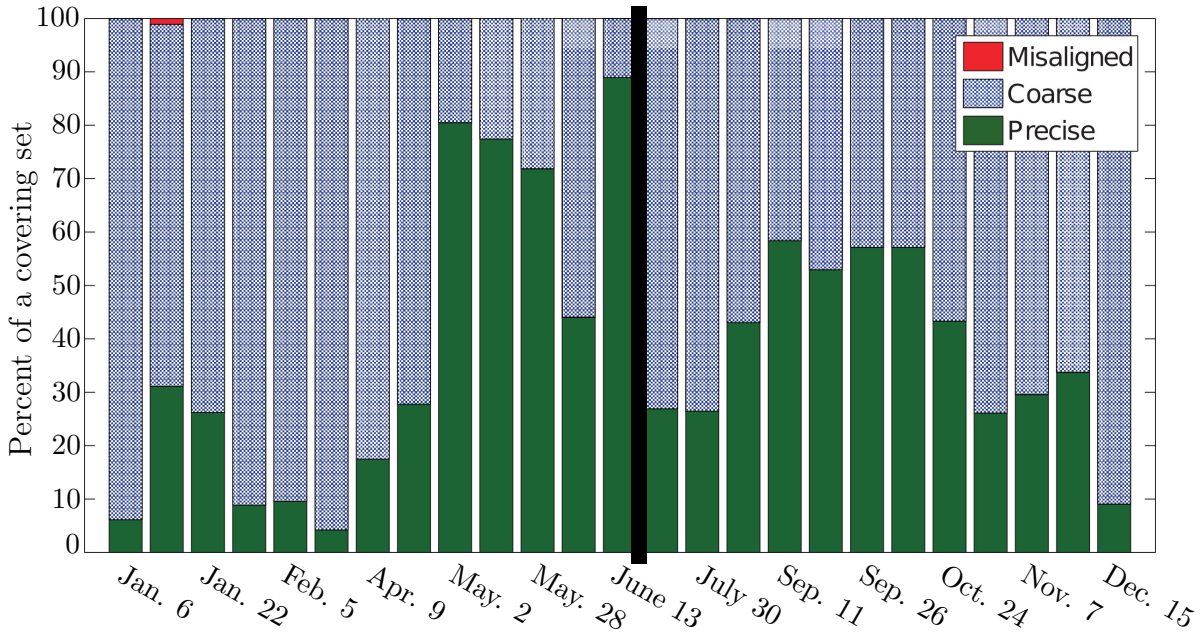

Figure 32: The alignment quality of the survey from June 25 with other surveys from 2014. The vertical bar denotes where the June 25 survey falls among these surveys.

many cases, however, a human would also have had difficulty identifying the precise alignment. For example, an image of a barren tree with a visible background may lack a direct correspondence with the fully leafed tree.

6.3.5 Alignment Quality Across Consecutive Surveys

Because some monitoring applications may only require image alignment between consecutive surveys, the alignment quality across ten consecutive surveys was evaluated. The ten surveys from 2014 were selected, which span a total time of 30 weeks, with one week the shortest interval of time between surveys and nine weeks the longest. Starting with the first survey as the reference survey, for each scene the corresponding image and the nearest one in the consecutive survey were aligned. As in the prior evaluation, a low-res search was applied to find the image pair that aligned best. A human evaluated the alignment quality. The process was repeated nine times to measure the alignment quality across the ten different surveys.

The results are shown in Fig. 33. A large number of precise alignments were found in all the comparisons, although some had more than others. The two cases with the

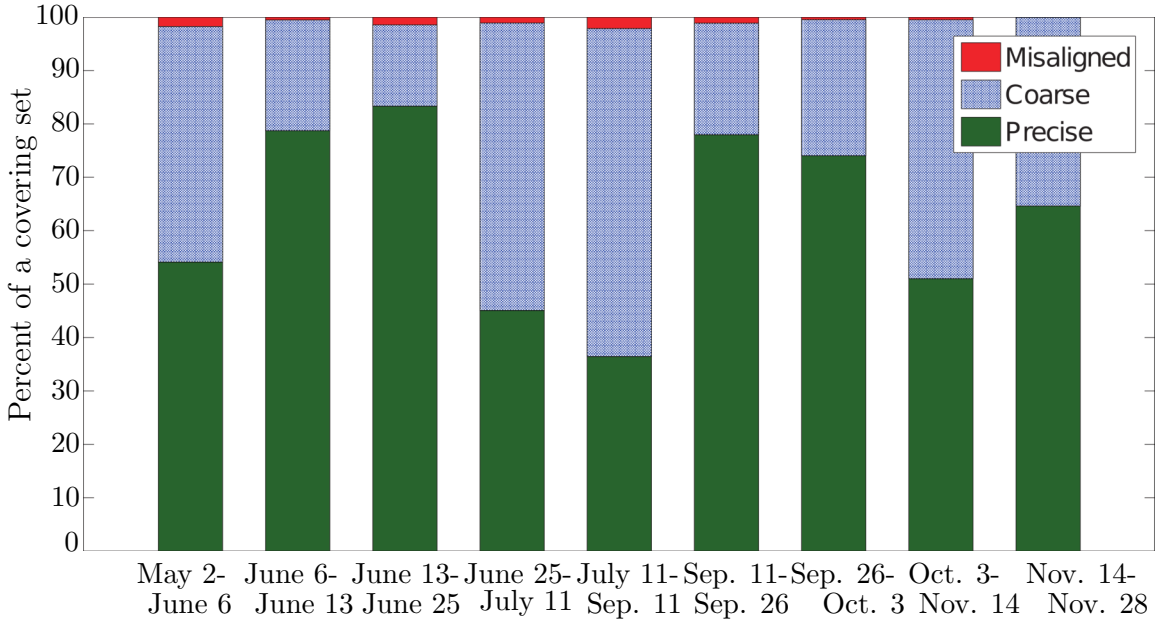

Figure 33: Alignment quality for comparisons of 10 different surveys. All 10 were captured in 2014.

fewest precise alignments involved a comparison with the July 11 survey, which was captured when the water level was higher. The upper half of many images in those cases were precisely aligned. Yet, because the perspective and the shoreline appearance significantly changed between surveys, SIFT Flow inaccurately extended the shore downward due to perceptual aliasing. Between the other surveys, the sun glare and the larger intervals of time between surveys (i.e., with more seasonal variation in the foliage) contributed to imprecise alignments.

6.3.6 Detected Changes

Several changes between surveys were found while labeling the alignment quality of each scene, which indicates that the aligned images and their combination in the flickering display facilitated change detection. Six examples are shown in Fig. 34. Five were found in precisely aligned image pairs; the "removed treetop" was identified in a coarsely aligned image pair. The flickering display was particularly useful for identifying small changes like the "cut branch". Although the large, floating tree was noticed while capturing a survey after a heavy rain (seen breaching the water in the

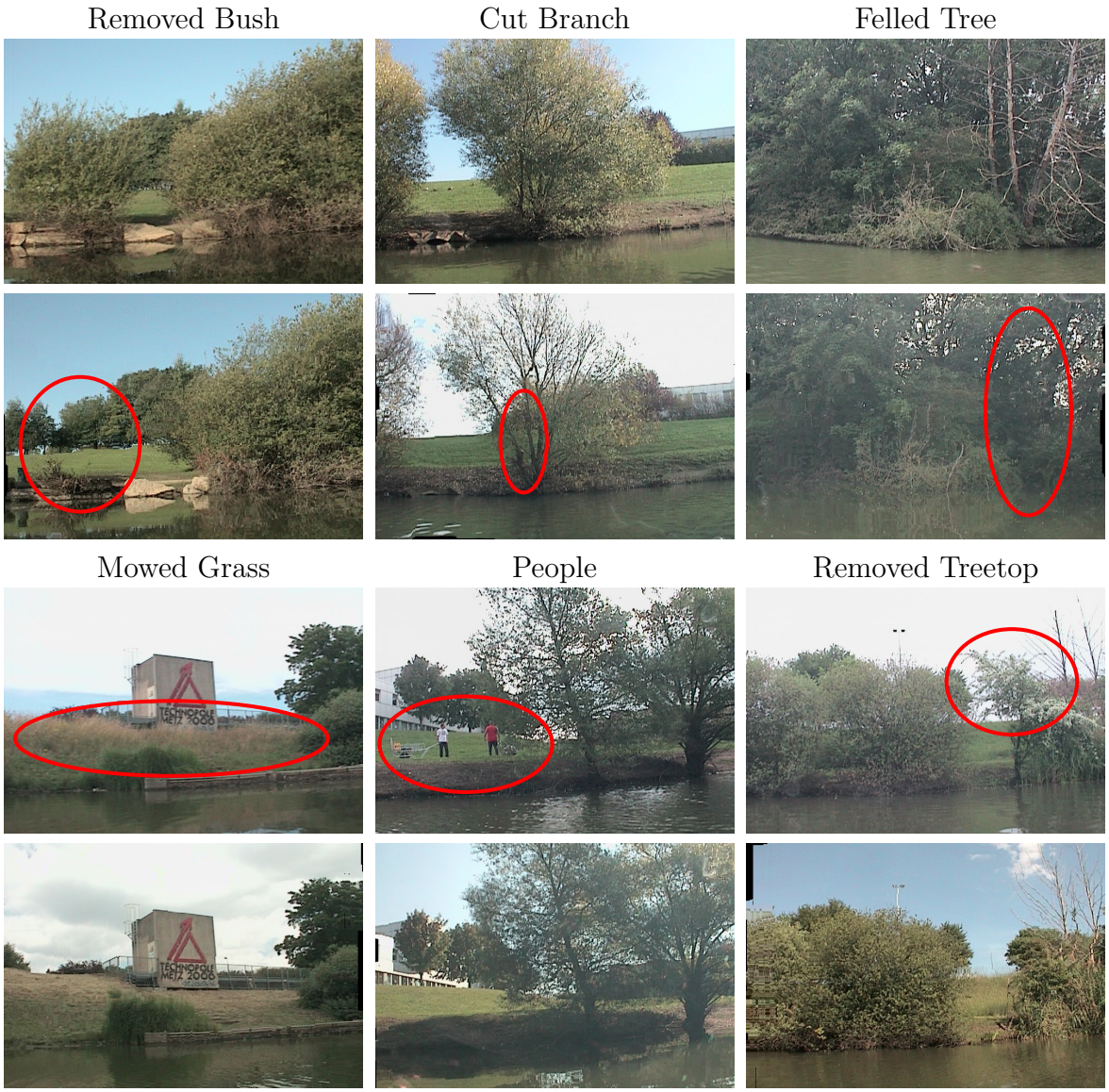

Figure 34: Six notable changes a human easily found while manually labeling the alignment quality between images from different surveys.

"sky and water" example of Fig. 35), it was not until this particular analysis that the prior location became known.

6.3.7 Robustness to Different Sources of Variation

The often high number of precise alignments between consecutive surveys indicates that SIFT Flow often had a high–level of robustness to the variation in appearance of a natural environment. Specific examples of well–aligned images across notable variation in appearance are shown in Fig. 35. Although SIFT Flow was not always robust to the variation in appearance, these examples show that it had robustness to at least some degree of illumination, glare, seasonal, and noise variation between image pairs. Of the six examples, the alignment across the most variation in appearance is perhaps the one labeled 'seasonal'. There, the image pair have different foliage, illumination, sky, water, shadows, and a globe reflection. Yet, results from Section 6.3.4 indicate that an alignment may have failed if the time interval between the pair was larger or if there was also sun glare.

6.3.8 Dense Correspondence Errors

The large number of coarse alignments between surveys indicates that many images did not align well; here the most frequent failure cases are provided. Six examples are shown in Fig. 36. Perhaps the most noticeable artifact is that SIFT Flow, as implemented, does not retain scene structures. This limitation has been observed in other image processing work as well (e.g., texture synthesis [92]). The "broken structure" example shows a tree trunk displaced from its top. In general, because each pixel is potentially warped differently than nearby pixels, the warp may be inconsistent across the image. Additionally, SIFT Flow often tried to align scene structures to noise (e.g., sun glare) and changes (e.g., a high water level) that obfuscated the scene. Out of all failure cases, most alignments were labeled "coarse" because they were translated versions of the same scenes.

Figure 35: Six different sources of noise across which SIFT Flow was robust.

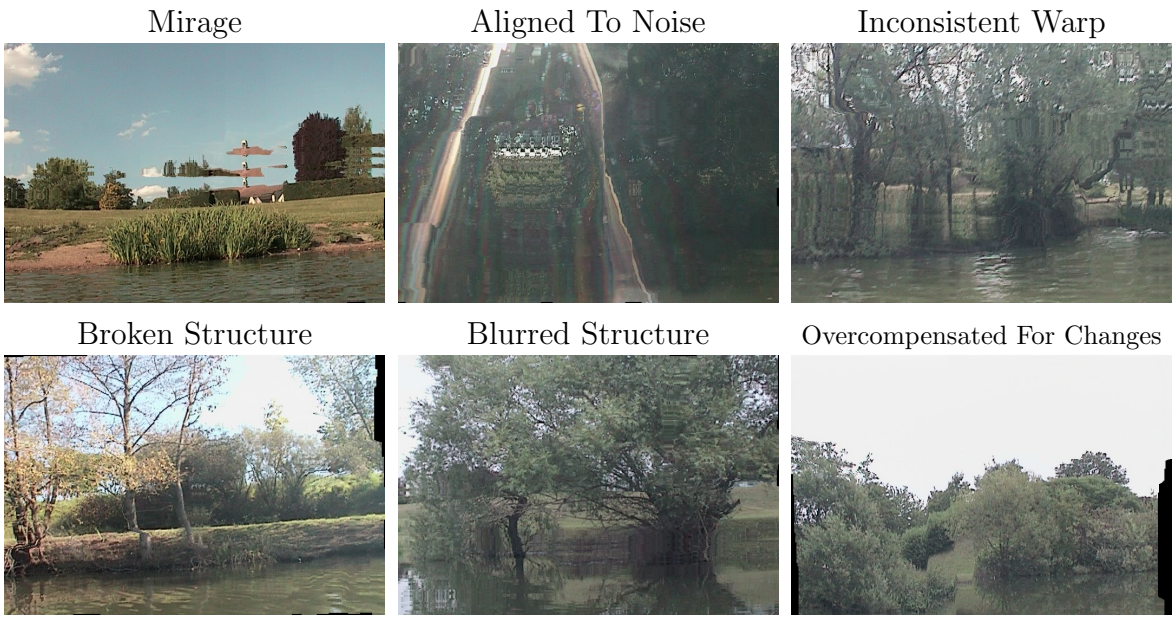

Figure 36: Six different alignment errors made by SIFT Flow.

6.4 Conclusion

This chapter made initial steps toward automating natural environment monitoring using a seeing, mobile machine. It found that while a side-by-side search for a single matching feature between two images of the same scene can be time consuming, existing algorithms can align entire images of a natural environment, and changes between them can be made readily apparent in a flickering display. Thus, rapid manual inspection inspection is possible when an image pair is aligned well. Of three methods, SIFT Flow was found to be the best suited for natural environment monitoring. SIFT Flow was unable to provide well–aligned images for every input, however.

The number of well–aligned images in the different scenarios demonstrates the suitability of dense correspondence for data association between images of a natural environment. Dense correspondence using SIFT Flow achieved accurate data association across marked variation in appearance. The proportion of precise alignments

decreased over time, which indicates that the Symphony Lake Dataset captures a representative real–world environment that is still a formidable challenge to be overcome. Indeed, there were many challenging image pairs that were not aligned well.

It is a hypothesis of this thesis that many of the coarsely aligned image pairs may be in reach of becoming precisely aligned. The formulation of SIFT Flow in this chapter mainly only relied on the appearance to align images. The 3D structure was used a mask to bias the alignment, but it was a use of 3D structure that was unable to short the variation in appearance of a natural environment. The 3D structure of a natural environment could yet lead to a much larger set of well–aligned images, particularly across seasons. The next chapter describes how 3D information may be integrated into the dense correspondence optimization to achieve image alignment across seasons.

CHAPTER VII

IMPROVED DENSE CORRESPONDENCE*

Dense correspondence power is highly dependent on the variation in appearance over time until the environment structure is used to anchor data association, which is shown in this chapter. A map–centric approach can, due to position–based correspondence, provide map point priors for robust data association. During a repeated survey, if the cameras are well–localized, the map from a different session can be projected onto the new images to provide position–based—independent of appearance—correspondence priors. Landmarks that keep the same position between surveys may be projected in this way. The projected landmarks will correspond to the same objects in the new images if a few conditions are satisfied (if there are objects that could occlude parts of the scene, the images may have to have been captured near more similar viewpoints). In a natural environment, trees, rocks, logs, and other objects that lack agency are prime examples of landmarks that could provide a map–centric correspondence. From the known positions of things in an environment, dense correspondence can begin with priors for data association, which are specified in a way that is largely independent of appearance given a consistent map and localized poses.

The question is how to integrate known 2D correspondences from reprojected map points into a dense correspondence optimization. Using map points as a mask to bias dense correspondence (Section 6.3.2) was insufficient because the scene structures were often violated. Instead, a set of prior 2D correspondences should hold two images in place where they are defined. Because reprojected map points would likely have reprojection error, the priors also should have flexibility at those locations

*This chapter is a paper that was presented at the 2016 British Machine Vision Conference (BMVC) [58].

proportionate to their error. And at the pixels without a prior, the priors in the local neighborhood should act as anchors to hold the rest of the image in place.

Accordingly, this chapter defines Reprojection Flow, a method that complements SIFT Flow using the spatial information in a map to provide data association priors across seasons. If multiple surveys are represented in one consistent map and set of poses, Reprojection Flow can 1) identify images of the same scene by the co-visibility of map points; and 2) initialize and constrain dense correspondence using reprojected map points as correspondence priors. To evaluate Reprojection Flow across a year of surveys, a single map was acquired for all the surveys by applying multi–session optimization to all the maps and poses and all the inter–session constraints between them. The maps and poses were given (from Chapter 4). The inter–session constraints had to be acquired. A search for inter–session constraints consisted of applying dense correspondence between near–time surveys. Correspondence consistency and epipolar constraints were adapted to dense correspondence to add robustness in this step. Results showed that Reprojection Flow consistently anchored dense correspondence across seasons, and in some cases resulted in time–lapses across an entire year of surveys. It was also found to provide some robustness to variation in viewpoint.

7.1 Reprojection Flow

Reprojection Flow [58] can provide map point correspondence priors between two images when one consistent 3D environment structure is available for all the surveys. Map points from one survey are reprojected onto the image from another to guide image alignment. Map–anchored dense correspondence may lead to more accurate image alignment when variation in appearance has accumulated. They may also help guide the correspondence search when perceptual aliasing is high. Given a localized pose, the *reprojection of map points* determines which viewpoint is selected (Sec. 7.1.1) and where dense correspondence is anchored (Sec. 7.1.2).

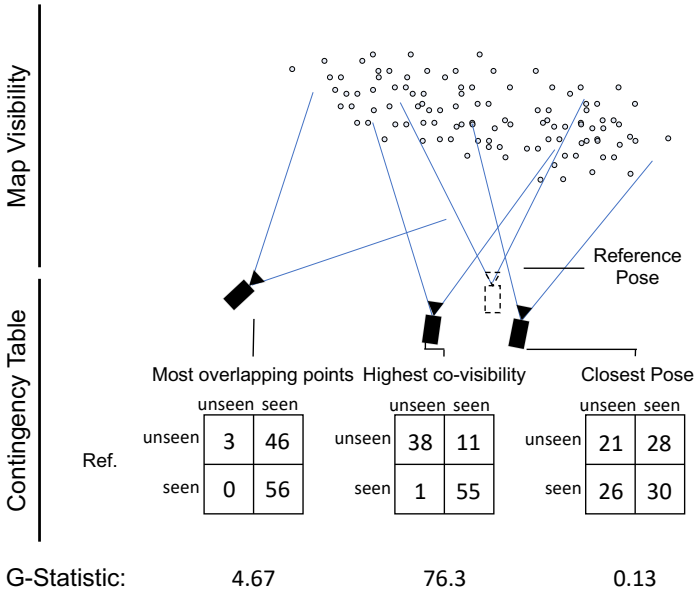

Figure 37: Viewpoint selection using the co-visibility of reprojected map points. The viewpoint with the most similar set of seen and unseen map points to a reference pose, as captured using a contingency table, has the highest co-visibility, and is the one for which the G-statistic is maximized.

7.1.1 Viewpoint Selection

To identify two images of the same scene, a co-visibility heuristic was defined in which two viewpoints capture the same scene if the same set of map points projects onto them (see Fig. 37). In contrast to an approach based on feature matching, this information can be independent of the time scale across which viewpoint selection is performed. Over a year, the appearance of a scene could change negligibly or completely. As shown in Chapter 5, if we relied on a visual feature descriptor (e.g., SIFT) to find the same scene, the difficulty of viewpoint selection could escalate with the variation in appearance of the environment. Given a consistent map and localized poses, however, the set of reprojected map points can provide information that is independent of appearance.

There are a number of ways to identify the same viewpoint in multiple surveys using a consistent map and localized poses. Two images capture the same scene if,

for example, given one camera pose, a nearby pose is pointing in a similar direction. If that heuristic was used, however, the scene contents would be unaccounted for—a distant scene may not be visible in both images. Alternatively, the number of map points that are visible in both images could be maximized. Yet, one image may capture a much larger area than the other. Instead, the image with the most similar viewpoint to a reference image views roughly the same set of map points.

Viewpoint selection utilizes *co-visibility*, a heuristic for maximizing the mutual information of reprojected map points (computed similarly to FAB-MAP from [27], but here based on point projection rather than to identify whether a place has been seen before using appearance features). Co-visibility is based on the property that a map point either projects onto an image or not. A viewpoint has high co-visibility to a reference image if the map points that project onto it also project onto the reference image, and the rest project outside of both images. Two viewpoints have low co-visibility if many map points project onto one image and not the other.

To calculate the co-visibility of two viewpoints, co-visibility statistics for all the map points are accumulated in a two–variable contingency table. The two rows of the table correspond to 'seen' and 'unseen' map points for one viewpoint, the two columns of the table for the other viewpoint, in the form:

	'unseen'	'seen'
'unseen'	N_{00}	N_{01}
'seen'	N_{10}	N_{11}

,

The co-visibility of two viewpoints is calculated using the G-statistic, a method from statistical analysis, which has been applied in robotics to, e.g., measure co-movement in [60], as:

$$G = 2\sum_{i=0}^{1}\sum_{j=0}^{1} N_{ij} \ln\left(\frac{N_{ij}(N_{00}+N_{01}+N_{10}+N_{11})}{(N_{0j}+N_{1j})(N_{i0}+N_{i1})}\right), \tag{2}$$

The co-visibility to a reference image is calculated for each candidate image of a survey using Eq. 2. The equation is maximized for the viewpoint with the highest co-visibility.

7.1.2 Map–Anchored Dense Correspondence

The set of reprojected map points that are co-visible in an image pair is used to anchor their alignment. Each map point specifies a precise correspondence between the images of two well–localized cameras when projected onto them (see Fig. 38). This *reprojection flow* directly constrains the pixels where the map points are reprojected. Indirectly, reprojected map points anchor the alignment consistency constraints, define the epipolar lines to which the other pixels are constrained (see Section 7.2.3), initialize their hypothesis spaces to average flow of the map points, and limit the range of their hypothesis spaces. Collectively, a dense correspondence may be nearly fully specified using Reprojection Flow before using any information about appearance.

Map point priors are added to image alignment using SIFT Flow to obtain the final dense correspondence. Although the appearance aids less in the alignment of images from opposite seasons, it can improve the alignment of images captured during similar time periods. Furthermore, the smoothed dense correspondence created by the MRF optimization may help reduce artifacts created by strong map point anchors. Those anchors are only correct up to the magnitude of their reprojection error. Thus, the alignment energy is as specified in Eq. 1 except for pixels that are map–anchored, whose alignment energy is:

$$
\begin{aligned}
E(\mathrm{w}) = {} & \mathrm{rf} + \mathrm{cyc} \\
& + \sum_{r \text{ adj. } to \ q} min(\alpha|u_q - u_r|, \mathrm{d}) + min(\alpha|v_q - v_r|, \mathrm{d}) \\
& + \sum_{q} \nu|u_q + v_q|
\end{aligned}
\tag{3}
$$

Because the data term of the energy function is a function of scene appearance, it

is replaced at the pixels where reprojected map points are specified. A suitable value for rf is calculated using the median of the data terms, t (from Eq. 1). That is,

$$\mathrm{rf} \propto (1 - \mathcal{N}(\kappa, \mathrm{s})) \times \mathrm{t}, \tag{4}$$

where κ is the pixel location of the reprojected point and s is the reprojection error divided by the image scaling factor. The cycle consistency, cyc, from Eq. 5 is still used. No alignment verification is performed when Reprojection Flow is used to guide the image alignment. Rather than project map points from all the surveys onto each image, only those from the two surveys that correspond to the two images are used, which limits the reprojection error to one direction.

Reprojected map points also define an initial hypothesis space at each pixel, which may help reduce perceptual aliasing for two reasons. First, the dense correspondence is initialized near the correct alignment (given an accurate pose transform between surveys), which is calculated as the average reprojection flow for all the landmarks of the reference image. Without Reprojection Flow, it is initialized to a zero-vector flow, which with the regularization term may create a bias in favor of the wrong alignment. Second, the hypothesis space is the L_∞ distance from the average flow of the reprojected map points. With a nearly correct initialization and a small hypothesis space, less information may be needed to pull the image into the correct alignment.

The accuracy of Reprojection Flow is based on the correctness of the map points and the camera poses. To minimize the effect of reprojection error, only the original set of landmarks that were viewed in the two to-be-aligned images are used (as opposed to all the map points at the same scene). Rather than use the 3D points, the tracked locations of KLT points were used, which limited the reprojection error for each point to one direction. Also, the outliers according to epipolar geometry were eliminated from the set. Beyond these constraints, some error was allowable because dense correspondence proceeded through a coarse-to-fine image pyramid.

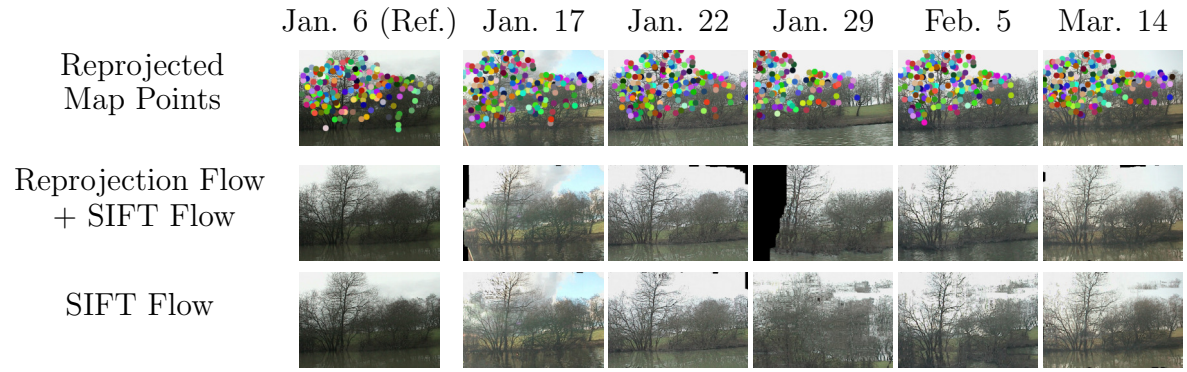

Figure 38: Map–anchored dense correspondence using Reprojection Flow for one scene from the Symphony Lake Dataset. Keypoint tracks from the reference survey (top left image) are shown reprojected onto images of the same scene from other surveys (top row). The locations of reprojected map points are the priors that anchor SIFT Flow to the final dense correspondence (middle row). Image alignment using the off-the-shelf version of SIFT Flow is provided for comparison (last row). Note that errors in the alignments produced using Reprojection Flow in this example occur in the areas of the images without reprojected map points (see e.g., the shoreline of the Jan. 29 image).

7.2 Acquiring One Map Of Multiple Sessions

Acquiring one consistent map and poses for a year of surveys in order to evaluate Reprojection Flow was a feat in and of itself. A search for inter–session constraints among all of the near–time sessions was undertaken (Section 7.2.1). When a pair of images were found to have correspondence between them, a modified version of dense correspondence was applied, in which alignment constraints were added for improved accuracy (Section 7.2.2).

7.2.1 Inter–Session Constraint Search

A large–scale search was run to find constraints between surveys in order to build one consistent map. Data association was only attempted between pairs of surveys where accurate constraints could be extracted. Due to the results of Chapter 5, the search was limited to pairs within three months of one another, and those from the beginning and the end of the year. Data association consisted of applying low–res SIFT Flow

to find pairs that aligned well. Inter–session constraints were acquired when full resolution dense correspondence proceeded. Inter–session constraints consisted of using the flow field to map keypoints between surveys (the mapping is explicated in Section 8.2.3.1), which resulted in inter–survey observations of landmarks. After the search, bundle adjustment was applied to all the maps, trajectories, and inter–session landmark observations to acquire one consistent map and poses.

Only the best alignments were used because many images had significant alignment noise. Image alignment quality was measured using alignment energy and match consistency (see Section 7.2.2) at the top layer of the image pyramid (i.e., low resolution). For an image I_a^j from survey j, a local (defined by GPS and compass limits) search for the image I_b^k from survey k that best matched I_a^j was found. If the alignment energy was < 1120000 (which mostly corresponded to high-quality alignments) and the consistency was $\geq 95\%$ (that is, 95% of pixels had error ≤ 1 pixel), the alignment qualified as one from which an inter–session 2D keypoint measurement could be extracted. Only the inliers according to epipolar geometry and match consistency were retained.

7.2.2 Alignment Consistency Constraints

Alignments were optimized with the help of an alignment consistency constraint. The consistency of an image alignment is measured using the alignment in the reverse direction. Because the dense correspondence is directional, that is, from one image to the other, a somewhat different one may be computed for the reverse direction. A different alignment may be likely for a highly self–similar scene. Matching the correspondence in the forward and the reverse directions may help reduce perceptual aliasing.

The alignment consistency is implemented as an iterative two–cycle correction in the low resolution stage of SIFT Flow. There, several iterations are inexpensive.

Pixels of the sift image S_b^k are first matched to pixels of the sift image S_a^j, then of S_a^j to S_b^k, and so on over at most 19 iterations. An odd number is used to end up at the forward flow, with which the next layer of the image alignment pyramid is initialized. Fewer than 19 iterations are performed if the consistency breaches 95% within one pixel. At that point the flow fields in both directions are consistent with one another.

Each iteration includes a modification to Eq. 1 to correct ambiguous correspondences. The data term of Eq. 1 is appended with the value

$$\text{cyc} = 16 \times ||\text{w}(q) - \text{w}^{prev}(q + \text{w}(q))||_2. \tag{5}$$

This term is the L_2 distance between the correspondence of the forward flows, w, and the flows of the previous iteration, w^{prev}, which are in the reverse direction. It is larger for pixel correspondences that diverge from consistency with the flow in the opposite direction. Its addition to the data term (rather than its multiplication to that) gradually pulls the forward and the reverse alignments into agreement.

7.2.3 Epipolar Constraints

The dense correspondence optimization also included an application of epipolar constraints. Corresponding points between two images of the same, static scene should fall on epipolar lines. The original implementation of SIFT Flow lacked epipolar constraints. Because SIFT Flow lacked that constraint, it was an opportunity to exploit more information for static dense correspondence.

Epipolar constraints guide the dense correspondence optimization after an initial set of correspondences are acquired. The very first set of correspondences is available after iteration 0 of the two–cycle consistency correction. Epipolar constraints are computed for each iteration thereafter using the previous flow field. They are also computed for successively larger layers of the image pyramid using the correspondences from the last.

Fundamental matrix estimation using RANSAC defines the epipolar constraint

for each pixel. The data term of Eq. 1 is multiplied with the value

$$\text{epi} \propto 1 - \mathcal{N}(\mu, \delta), \tag{6}$$

where μ is the L_2 distance to the epipolar line from $q + \text{w}(q)$ and $\delta = 2.5$. The data term is multiplied by the epipolar constraint in order to strongly influence the flow to obey epipolar geometry.

The use of two–cycle consistency and epipolar constraints changes Eq. 1 to:

$$\begin{aligned} E(\text{w}) = & \sum_{q} min(|S_b^k(q) - S_a^j(q + \text{w}(q))|_1, \text{t}) \times \text{epi} + \text{cyc} \\ & + \sum_{r \text{ adj. } to \ q} min(\alpha|u_q - u_r|, \text{d}) + min(\alpha|v_q - v_r|, \text{d}) \\ & + \sum_{q} \nu|u_q + v_q| \end{aligned} \tag{7}$$

7.3 Experiments

Reprojection Flow was evaluated in a series of experiments on data from 24 surveys of the Symphony Lake Dataset from 2014. A 100m long section was used to keep the map search and the multi–session optimization time tractable. Viewpoint selection is shown in Section 7.3.1 to find more similar scenes than image retrieval using visual slam (the best method for image retrieval from Chapter 5). Dense correspondence is shown to find consistent alignments across seasons (Section 7.3.2) and viewpoints (Section 7.3.3), far better than SIFT Flow.

7.3.1 Viewpoint Selection using Reprojection Flow

Viewpoint selection using Reprojection Flow was evaluated across all pairs of surveys, which resulted in a large–scale evaluation. Standard SIFT Flow was used as the image alignment method in this analysis. Images were aligned to identify the method with the lowest alignment energy. An image pair typically had lower alignment energy the better it captured the same scene. Thus, the alignment energy of viewpoint selection

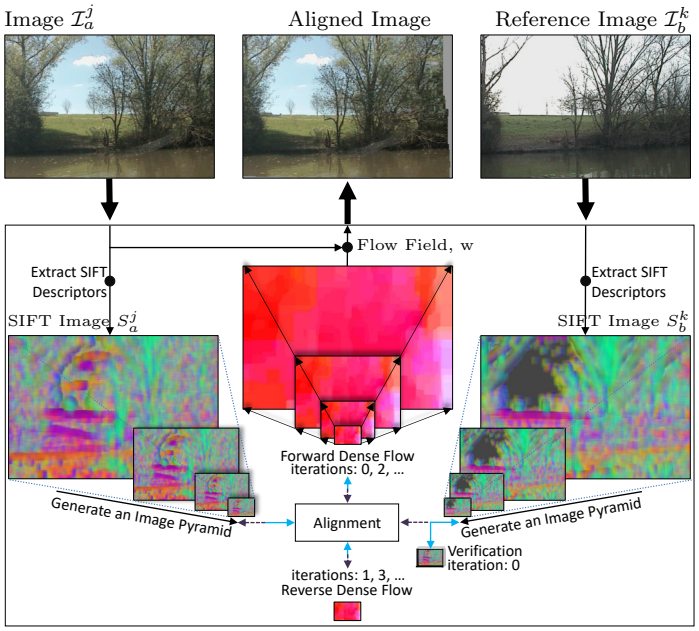

Figure 39: Depiction of image alignment using SIFT Flow plus alignment constraints. A SIFT Image is computed for each of the two input images. Each one is downsampled into an image pyramid with four layers. Image alignment proceeds from the top layer of the image pyramid down, with multiple iterations of alignment constraints applied at the top layer. An alignment is verified in iteration 0 (verification is described in Chapter 8, where the experiments first used it). To apply alignment consistency constraints, the forward flow field is computed in even iterations; the reverse flow field the odd iterations. Epipolar constraints are applied after iteration 0 and, unlike the alignment consistency, are also applied in the larger layers of the image pyramid.

using Reprojection Flow was better than that of image retrieval if, on average, the alignment energy of the image pairs it found was lower. In this case, the closest pose heuristic according to visual SLAM (on the map and trajectories from multi–session optimization) served as the image retrieval baseline. The difference in the alignment energy between the two approaches was calculated for each image pair and then accumulated for each pair of surveys. A total of 24*23=552 survey comparisons made up this analysis.

Figure 40 shows the result. Out of 552 survey comparisons, 547 reached lower alignment energies on average using Reprojection Flow. The co-visibility heuristic found image pairs that were, on average, closer to the same scenes. That was true

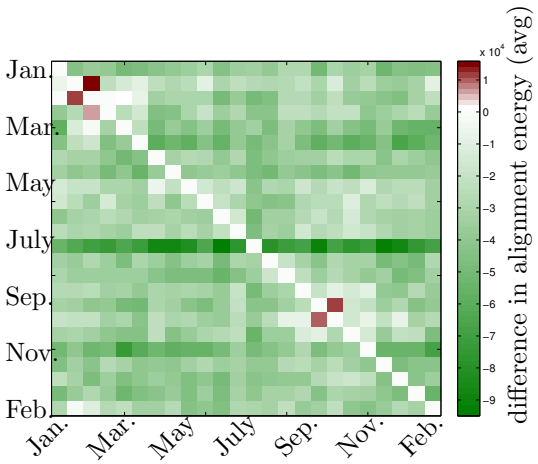

Figure 40: The average improvement in alignment energy of viewpoint selection using Reprojection Flow over viewpoint selection using the closest pose heuristic. Each square represents the result for aligning images from two surveys. A total of 24×23 survey comparisons make up this analysis. Lower is better. (*Best viewed in color.*)

across seasons. Finding closer images to the same scenes also simplified the task of dense correspondence because their contents had better correspondence.

7.3.2 Consistency of Dense Correspondence Across Seasons

Reprojection Flow was next tested on how well it could drive dense correspondence across seasons. Although hand–labeled 2D correspondences between an image pair are typically used to evaluate alignment quality (as in [175]), finding any correspondences can be difficult if the images capture a natural environment and span seasons. Instead, this evaluation relied on the prior set of image pairs that were found to align well from Section 7.2.1. For one such dense correspondence $I_a^j \rightarrow I_b^k$, the image I_c^l from survey l with the highest co-visibility to I_a^j was found, and then dense correspondence was performed for $I_a^j \rightarrow I_c^l$ and $I_b^k \rightarrow I_c^l$. This made a three-cycle between the images, i.e., $I_a^j \rightarrow I_b^k \rightarrow I_c^l$, which indicated alignment quality by how well the flows matched $I_c^l \rightarrow I_a^j$. The more points of the flow field closer to zero, the more consistent the dense correspondences. The average alignment consistency of Reprojection Flow was compared to that of SIFT Flow between surveys that spanned $j = 4$ to $6, 9$ to $11, \ldots,$ and 34 to 36 week differences.

Table 3: The average ratio of points within 15 pixels after three-cycle consistency. The values are high given that the analysis involves a three-cycle and 704x480 resolution images.

Number of Weeks Between Surveys	4-6	9-11	14-16	19-21	24-26	29-31	34-36
SIFT Flow	0.27	0.23	0.24	0.23	0.21	0.21	0.21
Reprojection Flow	0.31	0.30	0.31	0.31	0.29	0.30	0.32

Table 4: The average ratio of points within 15 pixels after three-cycle consistency, with added viewpoint variation.

Number of Weeks Between Surveys	4-6	9-11	14-16	19-21	24-26	29-31	34-36
SIFT Flow	0.19	0.15	0.16	0.17	0.14	0.15	0.16
Reprojection Flow	0.26	0.20	0.25	0.25	0.21	0.26	0.25

Table 3 shows the results. When SIFT Flow was guided by Reprojection Flow and alignment constraints, dense correspondence was significantly more consistent, on average, than SIFT Flow alone. Whereas SIFT Flow lost consistency across seasons, Reprojection Flow retained about the same level of performance. Reprojection Flow helped preserve the scene structures across seasons.

The difference in alignment quality for the two approaches is shown in Fig. 41, wherein the alignment quality was manually labeled for three different surveys across a sequence of eight different scenes. Broken scene structures were apparent more often with SIFT Flow. For example, the June images at section 375 show perceptual aliasing with reflections in the water, albeit more so with SIFT Flow. Reprojection Flow lost the scene structure near the shoreline of that image due to the error of the map points there. The more accurate map points at section 325, however, kept the scene structures in place during the alignment. Similarly, the other images marked green better retained the foreground structure.

7.3.3 Consistency of Dense Correspondence Across Seasons and Viewpoints

Because Reprojection Flow centers the hypothesis space around the correct alignment, it can make dense correspondence more robust to variation in viewpoint. To evaluate

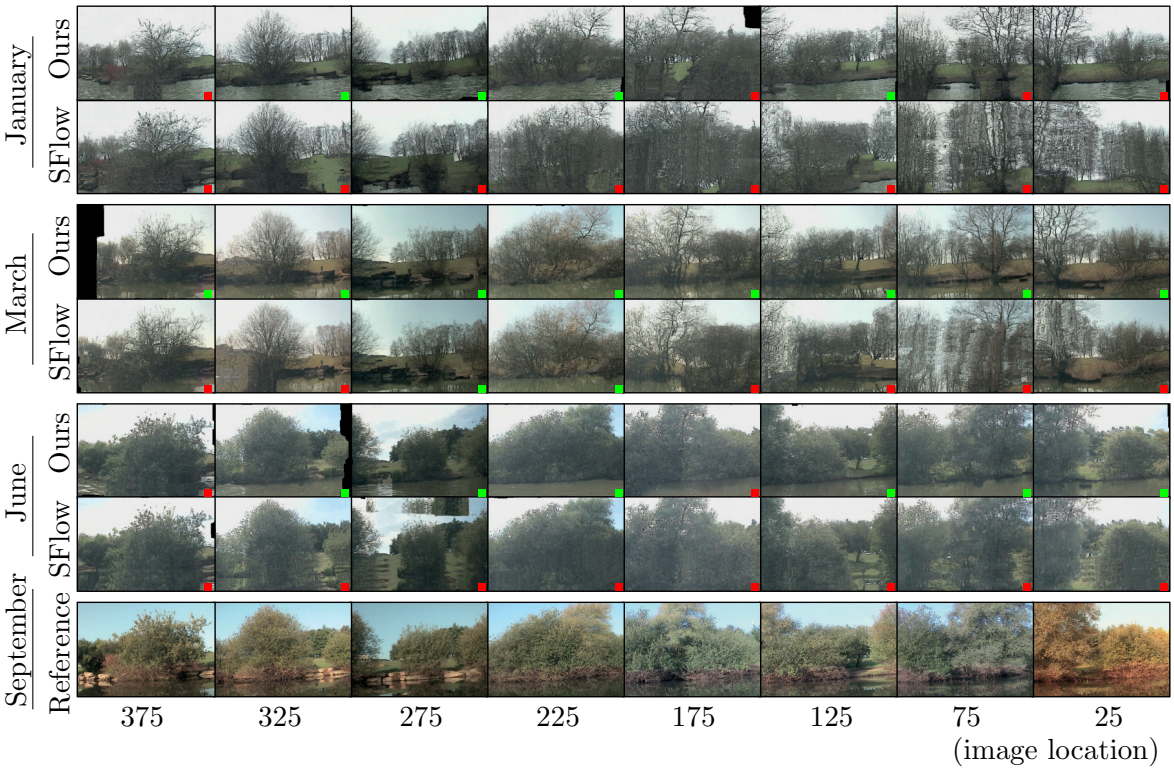

Figure 41: Images from a September survey shown here aligned with images from a January, a March, and a June surveys using Reprojection Flow and SIFT Flow. Green and red flags indicate alignment quality as manually labeled by a human. The foreground mostly aligned well in the images marked green whereas significant artifacts appeared in the images marked red.

this, the same test as in Section 7.3.2 was performed, except using images offset from I_c^l. An offset of $d = c - 10$ was used, which corresponded to a displacement of the scene in the image by roughly 25-50% of the image width. The result is shown in Table 4. The improvement over SIFT Flow was retained.

7.4 Conclusion

This chapter showed that the positions of static objects in an environment can provide accurate priors for dense correspondence across seasons. With Reprojection Flow, a localized observer knows what scene it is viewing. It also knows how the scene corresponds to views of the same place it acquired in the past. Map points reprojected onto images to provide this appearance-invariant information. Alignment constraints helped to maximize it.

Knowing the rough location of the correct alignment in the image is powerful, for robustness to variation in both appearance and viewpoint. The hypothesis space could be made much tighter when it was centered around the correct alignment. This advantage means that in the cases when few image descriptors matched well, the best match was still often the correct one. SIFT Flow used a large hypothesis space to compensate for the fact that it was not centered at the correct one. With the same parameters as used in Chapter 6, its hypothesis space was large enough to compensate for correspondences up to 50% of the image width. It may have been made larger, but that would have added many more candidate matches, which could have lead to significant artifacts in the image due to perceptual aliasing.

The next chapter sought to confirm and extend the results of this chapter by improving on the map acquisition process. In this chapter, map consistency highly depends on the accuracy of the initial set used to create inter–session correspondences. Alignment constraints were added in this chapter for robustness, but more could be done to filter outliers. Additinally, the optimization did not scale well; the optimization was performed on a machine with 192 GB of RAM. Both limitations are addressed in the next chapter.

CHAPTER VIII

TRANSFORMING MULTIPLE VISUAL SURVEYS INTO TIME–LAPSES*

A map–centric approach to data association can be used to establish dense correspondence across seasons in a natural environment, but the method is highly dependent on how the consistent map is acquired. Chapter 7 showed that dense correspondence can hold image alignment close to the map point priors. But time–lapses were only produced at a few scenes of the environment (a 100m section of the full 1.3km perimeter) where a map could be formed, and only where it was consistent. Although Reprojection Flow was able to establish correspondence across seasons, a better approach was needed to acquire a consistent map of a natural environment to show its full power.

Many of the challenges to acquiring one consistent map from multiple surveys of a natural environment were naively addressed in Section 7.2.1. The automatic identification of good image alignments needed improvement. As dense correspondence is applied between surveys increasingly separated in time, inaccurate inter–session constraints will start to be acquired, which reduce map consistency. The evaluations in e.g., Section 6.3.3 showed that a human could identify well–aligned images, and then heuristics were defined in Section 7.2.1 to automate the process. A well–aligned image was one whose dense correspondence energy was below a threshold and whose alignment consistency was high; two constraints that worked for a small section of shore, but that do not correspond to a well–aligned image pair in general. Expanding to the full environment (and beyond, to other environments) would lead to many false

*This chapter is a paper that is under review for publication in the International Journal of Robotics Research (IJRR) [55].

positives and negatives. A more general procedure was needed to delineate accurate inter–session constraints from outliers.

Mapping the environment anew in each survey leads to the need for a scalable optimization as more surveys are added and more of the environment is considered. More surveys and larger environments creates a larger problem for optimization, which has more variables and more inter–session constraints between sessions. Section 7.2.1 used one inter–session landmark observation for each landmark observed in a well–aligned image (for up to $300 \times 2 = 600$ constraints). It also left the representation of all the constraints among all the surveys undivided, which meant that optimization was over an extremely large number of variables and constraints. A research–grade machine was needed to solve the optimization problem, which took 192 MB of RAM and approx. a day of computation time. Using, instead, a pose transform for a set of landmark observations, which provides one constraint between an image pair (instead of 600) and subdividing the problem to optimize it using a divide–and–conquer approach (as in, e.g., [126]) could lead to suitable map–acquisition approach for environment monitoring.

This chapter introduces a framework to help transform multiple visual surveys of a natural environment into time–lapses. A consistent map is built from a set of inlier loop–closures with a divide–and–conquer optimization over all the surveys. After single–session SLAM is applied to all the surveys (Chapter 4), a search for inter–session loop closures is run between each pair of surveys within three months of each other (See Fig. 42). A pipeline of verification and geometric constraints are applied during the inter–session loop closure search, which replace prior, non-general heuristics. The Reprojection Flow algorithm of Chapter 7 is used both during the inter–session loop closure search and for forming time–lapses once a consistent map has been acquired. Multi–session optimization is applied to all the surveys and inter–session loop closures, as in Chapter 7, but now reaches a consistent map by

a dividing–and–conquering the optimization problem over several iterations, which includes a filter for outlier loop–closures.

The evaluation of this framework on the Symphony Lake Dataset led to year–long time–lapses of many different scenes. In comparison to another approach based on using ICP plus a homography, this framework produced more and better quality alignments. With many scenes of the 1.3 km environment consistently aligning well in random image pairs, the accuracy of 100 time-lapses across 37 surveys captured in a year was evaluated. Approximately one third had at least 20 (out of usually 33) well-aligned images, which spanned all four seasons. With promising results, the pose error of misaligned image pairs was evaluated, which showed that improving map consistency could lead to even better results.

8.1 Related Work

Transforming visual surveys into time–lapses using a consistent map touches on many challenging areas of data association. Of the areas not covered in Chapter 2, this chapter touches on scalability for visual data association (Sec. 8.1.1), scalability for backend optimization (Sec. 8.1.2), and robustness for backend optimization (Section 8.1.3).

8.1.1 Scalable Visual Data Association

Although offline processing of surveys does not need real–time scalability in visual data association for storage and retrieval, scalability can become a bottleneck if they are intractable [151]. Pose priors and the co-visibility of map points are used in this dissertation (Sec. 7.1.1) to mitigate the bottleneck. Related work has also applied co-visiblity heuristics, namely for appearance–based localization. Because visual features typically co-occur with other visual features on the same objects, appearance–based matching can exploit the distribution of features that may be observed there [27]. An image is likely to match a query image if its visual features are highly co-occurring

with those of the query image, as measured using mutual information in a Chow Liu tree. In the formulation of covisibility graphs [80, 155], a query image is localized to the node and its neighbors with the highest visual word frequency–inverse document frequency.

A number of other approaches are formulated for map maintenance to keep localization time small [121, 38]. [148] store a descriptor for each 3D map point and localize as soon as enough correspondences are found. [38] acquire a summary map, of which landmarks are those likely to be matched in future runs and trajectories are those that capture novel structure. [99] prioritize the sessions to which localization is attempted. In this chapter, the map of each survey is left as-is. But viewpoint selection avoids any descriptor comparison when the relative poses between sessions is known. Before a localization is acquired, the image retrieval in Sec. 8.2.1 gets a boost in scalability due to its low–resolution image alignment, similar to the idea to use compact image templates to keep image comparison fast [118, 4].

8.1.2 Scalable Backend Optimization

Scalability in backend optimization is achieved in a number of ways [14]. The multi-session optimization presented in Sec. 8.4 breaks the optimization into subgraphs, motivated by the scalability of [126] and [113]. The divide-and-conquer approach of [126] was extended to multi-session SLAM by [113]. One anchor variable is defined between each pair of sessions to represent the pose transform between them. The formulation is best if the poses are locally well-constrained. The approach in this chapter does not use anchor variables, but instead optimizes over all the loop closures between multiple sessions. Although several papers have shown scalability for real–time operation by keeping the pose graph small [16, 77], or the optimization over it small [150, 81], neither the implementation of single–session SLAM (Sec. 4.4) nor the implementation of multi–session optimization (Sec. 8.4) are used for real-time

operation.

8.1.3 Robust Backend Optimization

The backend optimization may also have to be robust to outliers in data association due to the high possibility of bad loop closures [162, 43]. Robustness can be explicitly added in an optimization over loop closure constraints. The multi–session optimization in Sec. 8.4 employs expectation maximization, which has been used to eliminate outlier loop closures of distributed mapping algorithms [35, 72]. Other techniques have optimized for the likelihood of a tree of loop closure candidates [43], optimize for clusters of inliers that share consensus with the odometry [96], and optimize for graph consistency [52].

A number of approaches also filter outliers by encoding the uncertainty within the factor graph. Switchable constraints are binary variables that can be added for each loop–closure to filter poor ones [157], which have been extended so that outliers are dynamically rejected [1], and to account for multiple hypotheses of variables [130]. [19] implicitly modeled switchable constraints in smart factors, which are an abstraction for the support variables of an optimization problem. They replace the support variables to provide a reduced set of constraints on the set of target variables for optimization. Smart projection factors were included in the single–session SLAM problem of Chapter 4. [18] gain robustness to spurious measurements by modeling constraints using noise distributions that account for large errors.

8.2 Inter–Session Loop–Closure (ISLC) Search

After multiple surveys are collected and optimized (using single–session SLAM from Chapter 4), connections are acquired between them during the inter–session loop closure (ISLC) search (see Fig. 42). An ISLC connects two surveys with a pose transform, which is extracted from a pair of aligned images (a formal definition is given in Sec. 8.2.3.4). As in Chapter 7, the dense correspondence of two images defines their

alignment (Sec. 6). Although dense correspondence provides a potentially more accurate alignment function (compared to e.g., a homography computed from local image correspondences), and may short a large degree of variation in appearance, it can still fail to provide *any* accurate correspondences. Therefore, its power is supplemented with a pipeline of constraints to help filter and circumvent errors—from the use of alignment constraints (Sec. 7.2.2), to outlier removal (Sec. 8.2.2 and 8.2.3.1), to the localization setup (Sec. 8.2.3.2 and 8.2.3.3), and finally to loop closure verification (Sec. 8.2.3.4). Section 8.3 shows how Reprojection Flow can be added to this pipeline to provide map point priors between two surveys once a loop–closure is verified.

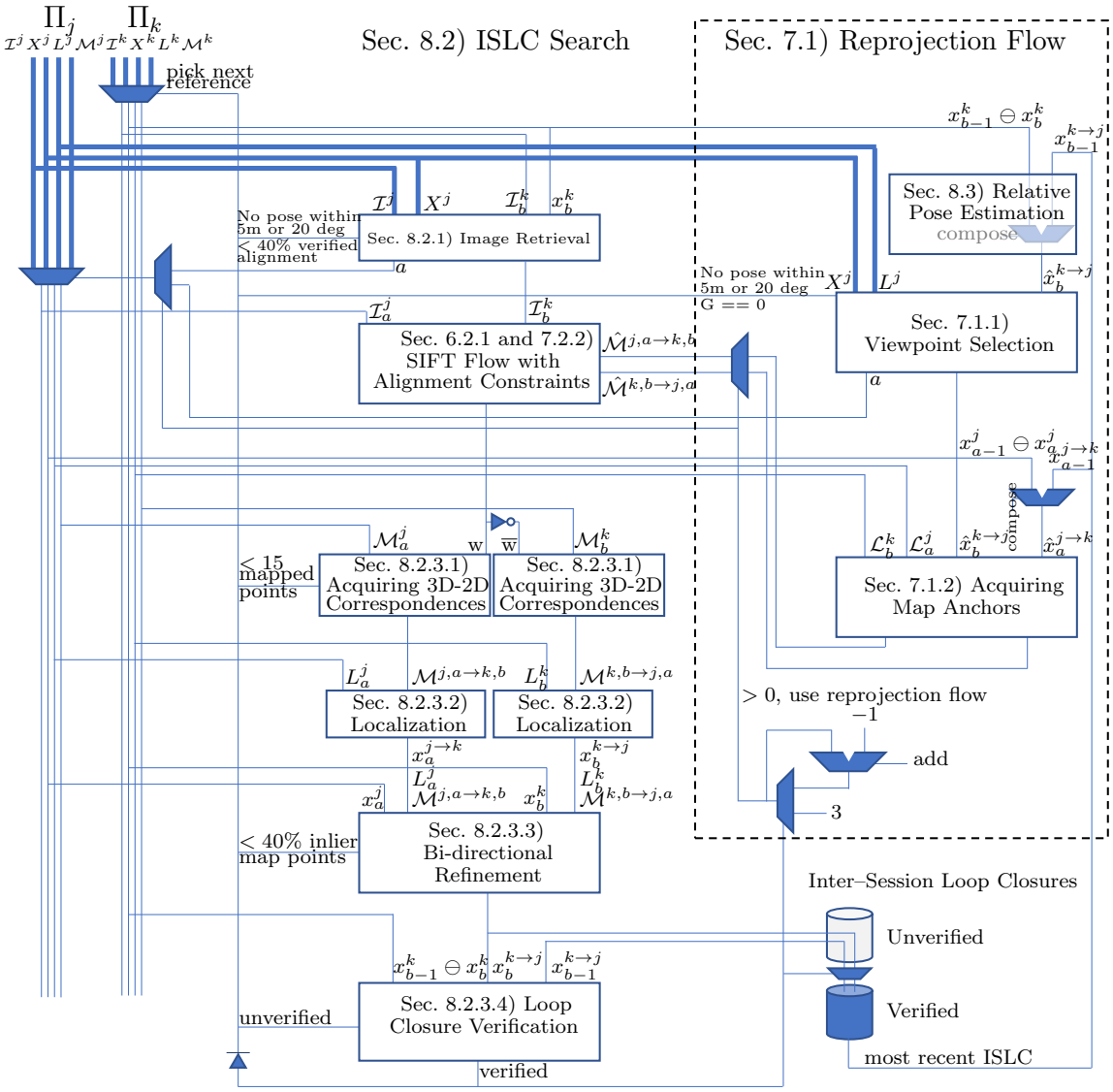

Figure 42: Visual data association between **(left)** two surveys using **(middle)** inter–session loop closure (ISLC) search (Sec. 8.2) and **(right)** Reprojection Flow when an ISLC is acquired (Sec. 7.1). Reprojection Flow is used up to three times without success before it is disabled. The logic here specifies the search with Reprojection Flow in the forward direction, but it is also used in the reverse direction. Also, the most recent ISLC is not necessarily between the times $a-1$ and $b-1$. See the text for details.

8.2.1 Image Retrieval

Data association between two surveys, j and k, begins with image retrieval, which seeks the best candidate image, $\mathcal{I}_a^j$, from survey j at time a for data association to a reference image, $\mathcal{I}_b^k$, from survey k at time b. It is implemented in this chapter to reduce computations of full–resolution dense correspondence (which can be computationally expensive) that are likely inaccurate. The search first identifies the poses $\hat{x}_{\text{I}}^j..\hat{x}_{\text{II}}^j$, from survey j near the pose $\hat{x}_b^k$. For the Symphony Lake Dataset, nearby poses are those within 5 m and 20 degrees of $\hat{x}_b^k$. The search then tests the corresponding image candidates, $I_{\text{I}}^j..I_{\text{II}}^j$, for alignment. A low–resolution dense correspondence (a good indicator of the full–resolution correspondence quality) is computed for each pair $\{\mathcal{I}_b^k, I_\gamma^j\}_{\gamma=\text{I}}^{\text{II}}$. This search is parallelized. An image $\mathcal{I}_a^j$ is found if at least one of $\{\mathcal{I}_b^k, I_\gamma^j\}_{\gamma=\text{I}}^{\text{II}}$ has a verified alignment (as defined in the next section). The one whose alignment is most verified with the reference image is the one that is returned.

8.2.2 Verified Dense Correspondence

Verification was added to the SIFT Flow framework to help identify whether an alignment may be informative (see Fig. 39). An informative alignment may be robust to noise, which is a property that can be *verified*. Without verification, the alignment process is terminated. A verified alignment can likely be, in turn, optimized. *Alignment consistency constraints* (Sec. 7.2.2) and *epipolar constraints* (Sec. 7.2.3) were used to optimize verified alignments within the SIFT Flow framework.

The robustness of an alignment is tested immediately after obtaining the low–resolution correspondence from the top of SIFT Flow's alignment pyramid. Noise is added to one image and the image pair is aligned a second time to test how much of the dense correspondence is retained. This is similar to the idea of 'adversarial perturbation,' wherein noise is added to an input image to test the robustness of a neural network (see, e.g., [37]). The second alignment verifies the first one if a large

percentage of the two dense correspondences match, which indicates that information may have been acquired [160, 154]. For surveys from the Symphony Lake Dataset, a dense correspondence of two images was verified if, after shifting one of the images up and to the right three pixels and then re-aligning them, at least 40% of the second dense correspondence matched the first (Appendix A describes how these values were reached). Note that, as implemented, alignment verification happened as part of image retrieval (rather than during the full–resolution dense correspondence).

8.2.3 An ISLC From a Flow Field

An accurate flow field, w, between images $\mathcal{I}_a^j$ and $\mathcal{I}_b^k$ specifies the dense correspondence of one image to another and is used to solve the PnP problem. The PnP problem is that of finding the pose of a camera from a set of 3D-2D correspondences. The result is a localized pose, which can become an inter–session loop closure constraint. There are four steps in the process: **Sec. 8.2.3.1)** acquiring 3D-2D correspondences from the flow field and the landmarks of each survey; **Sec. 8.2.3.2)** inter–session localization using the 3D-2D correspondences; **Sec. 8.2.3.3)** dual refinement of the two localized poses; and **Sec. 8.2.3.4)** a one-step loop–closure verification.

8.2.3.1 Acquiring 3D-2D Correspondences

Each landmark from one image is mapped to a 2D coordinate of the other using the flow field, which results in two sets of 3D-2D correspondences (one for each direction) (see Fig. 43). More formally, the 2D coordinates of landmarks $\mathcal{M}_a^j = \{m_\psi^{j,a}\}_{\psi=1}^{n_{j,a}}$ that were observed in $\mathcal{I}_a^j$ are mapped to pixels of $\mathcal{I}_b^k$ as $\mathrm{w}(m_\psi^{j,a}) \rightarrow m_\psi^{j,a\rightarrow k,b}$. The flipped flow, $\overline{\mathrm{w}}$, which is obtained with reverse lookup, provides $\overline{\mathrm{w}}(m_\varphi^{k,b}) \rightarrow m_\varphi^{k,b\rightarrow j,a}$, for $\varphi \in 1..n_{k,b}$.

The mapping of landmarks through a flow field provides an approximation, which is further refined using epipolar constraints. Landmarks are, in this framework, assumed to lack feature descriptors for matching across surveys. This framework's

substitute is a mapping through the flow field, a result that may have slightly diverged from the true landmark locations. Thus, after mapping all the landmarks, the subset that satisfies epipolar geometry are retained. The correspondences are discarded if there are fewer than 15, which typically occurs when the flow misaligns scene structures.

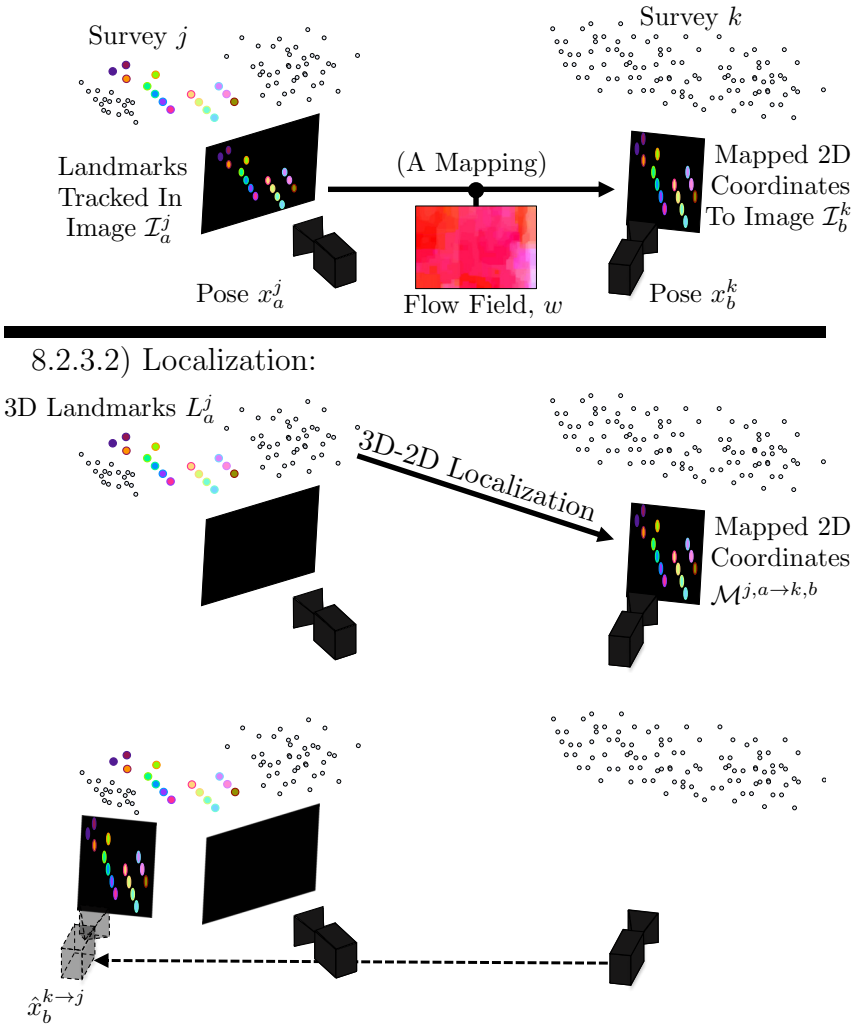

Figure 43: Localization to a prior survey after using a flow field to acquire 3D-2D correspondences. 8.2.3.1) A flow field defines a mapping from pixels of one image to another, with which the landmarks L_a^j seen in $\mathcal{I}_a^j$ are mapped to pixels $\mathcal{M}^{j,a\to k,b}$ of $\mathcal{I}_b^k$. 8.2.3.2) Localization proceeds as bundle adjustment using 100 iterations of RANSAC, each with 15 random 3D-2D point correspondences of the tuple $(L_a^j, \mathcal{M}^{j,a\to k,b})$. The result is the localized pose x_b^k in survey j, i.e., $x_b^{k\to j}$.

8.2.3.2 Localization

A set of 3D-2D correspondences is used to localize the camera pose of one survey to the other survey, as shown in Fig. 43, which is the perspective-n-point (PnP) problem. The 6D pose that corresponds to the mapped 2D image coordinates, x_a^j or x_b^k, is localized to the survey for which the 3D points are given, k or j, respectively. Because there are two directions of 3D-2D correspondences, a dual localization problem is formulated in which both camera poses are localized together.

Localization is performed by applying bundle adjustment to a factor graph that represents a random sample of 3D-2D correspondences. Points of the tuple $(L_a^j, \mathcal{M}^{j,a\to k,b})$ are the 3D-2D correspondences used to compute $x_b^{k\to j}$, pose x_b^k in survey j (see Fig. 43). A set of 15 correspondences is randomly sampled from the tuple (ongoing work has replaced this part with P3P, which uses three correspondences). A factor graph is created with one node for the pose and one localization factor for each of the 15 correspondences (a localization factor is a projection factor with a constant landmark and its implementation here is due to [6]). The application of bundle adjustment minimizes the reprojection error (the error between the tracked pixel location of a landmark and its reprojected location) of the factors.

The estimate of $x_b^{k\to j}$ is refined in multiple iterations of RANSAC. The new estimate in each iteration is graded according to the number of inlier 3D-2D correspondences. Inliers have a reprojection error of less than 6.0 pixels. A better value for $x_b^{k\to j}$ is acquired if the estimate has more inliers. RANSAC is stopped after 100 iterations (ongoing work has made this parameter adaptive in order to use fewer iterations if there are more inliers).

The same procedure is applied to $(L_b^k, \mathcal{M}^{k,b\to j,a})$ to get $x_a^{j\to k}$.

8.2.3.3 Bi-Directional Refinement

The RANSAC procedure provides a close initial estimate of $x_a^{j\to k}$ and of $x_b^{k\to j}$, which are further refined using an expectation–maximization bi-directional bundle adjustment. Because the pair of tuples correspond to the same flow, the estimate of $x_a^{j\to k}$ is tied to the estimate of $x_b^{k\to j}$. If the two estimates are left as-is, optimized separately, the difference between them could skew later image and survey alignments. Bi-directional bundle adjustment may pull them into closer agreement. Additionally, in contrast to the RANSAC step, all the inlier 3D-2D correspondences are used in each iteration of expectation maximization.

A two–variable factor graph that corresponds to $x_a^{j\to k}$ and $x_b^{k\to j}$ is used to represent the bi-directional bundle adjustment. A factor is added to represent the constraint that

$$x_b^{k\to j} = x_a^j \oplus (x_a^{j\to k} \ominus x_b^k), \tag{8}$$

where $\oplus$ is the compose operation in the SE(3) lie group, and $\ominus$ the between operation. A localization factor is also added for each inlier 3D-2D correspondence. The poses $x_b^{k\to j}$ and $x_a^{j\to k}$ and the reprojection error for each 3D-2D correspondence are updated after each iteration, which can change the set of inliers. The optimization is terminated when the number of inliers stops changing or after 15 iterations. The result is discarded if fewer than 40% of the 3D-2D correspondences are inliers.

8.2.3.4 Loop Closure Verification

An inter–session loop closure is acquired if the localized pose $x_a^{j\to k}$ passes a one-step verification using the nearest localized pose and the known change in pose (see Fig. 44 and e.g. [96]). This verification step is similar in principle to alignment verification (Sec. 8.2.2): A localized pose that is an informative one may be robust to noise, which is a property that can be verified. Once verified, the localized pose index, (j, a), the

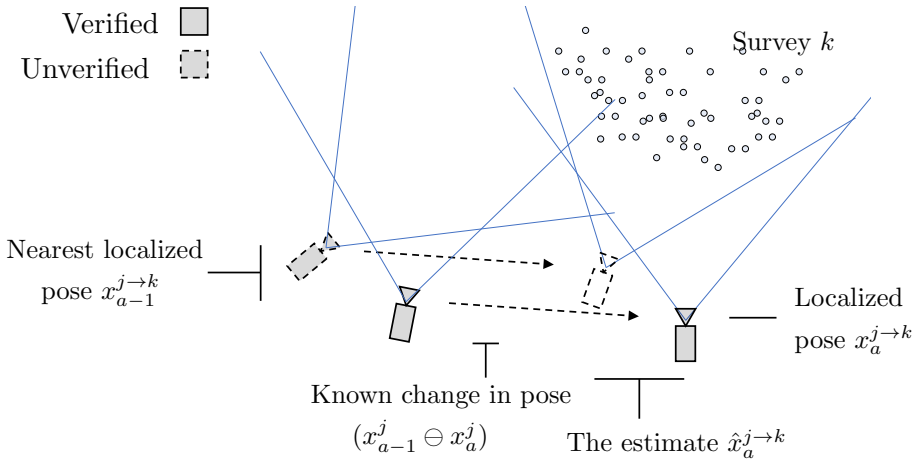

Figure 44: One–step verification of the localized pose $x_a^{j\to k}$ using the nearest localized pose, e.g., suppose $x_{a-1}^{j\to k}$, and the known change in pose, $(x_{a-1}^j \ominus x_a^j)$. The map points observed at x_b^k are projected onto the localized pose, $x_a^{j\to k}$, and the estimate, $\hat{x}_a^{j\to k}$. **solid)** The loop–closure is verified (at which point it is added to the set of ISLCs) if the map points project onto nearby pixels of both images ($x_{a-1}^{j\to k}$ is consistent with $x_a^{j\to k}$). **dotted)** The loop–closure remains unverified if the map points project onto distant pixels ($x_{a-1}^{j\to k}$ is inconsistent with $x_a^{j\to k}$).

reference pose index, (k, b), and the transform between their poses, $x_a^j \ominus x_b^{k\to j}$, are composed into an ISLC

$$(j, a, k, b, x_a^j \ominus x_b^{k\to j}), \tag{9}$$

which is added to the set of ISLCs, H^j, for survey j that are used for multi-session optimization (Sec. 8.4). An ISLC that corresponds to $x_b^{k\to j}$ is also added to the set H^k for survey k, which simplifies applying the same constraint to both surveys.

The pose $x_a^{j\to k}$ is verified using a set of 3D points, the known change in pose, and the localized pose that is nearest in the sequence, e.g. suppose the one captured at time $a-1$, i.e. $x_{a-1}^{j\to k}$, which may not yet have been verified itself. An estimate $\hat{x}_a^{j\to k}$ is computed using the known change in pose between x_{a-1}^j and x_a^j as

$$\hat{x}_a^{j\to k} = x_{a-1}^{j\to k} \oplus (x_{a-1}^j \ominus x_a^j). \tag{10}$$

The 3D landmarks observed at x_b^k are projected onto both $x_a^{j\to k}$ and $\hat{x}_a^{j\to k}$. Both localized poses $x_a^{j\to k}$ and $x_{a-1}^{j\to k}$ are verified if at least 25% of the points project onto

both images and their average reprojection error is less than 6.0 pixels.

8.3 Reprojection Flow Within The ISLC Search

Reprojection Flow can provide map point correspondence priors between two images when the pose transforms between them are known. This includes times during the ISLC search when the relative poses between two surveys can be accurately estimated. In that case, Reprojection Flow is used in the same way: After estimating the localization of a pose, the reprojection of map points determines which viewpoint is selected and where dense correspondence is anchored.

Relative pose estimation is the step of estimating the next localized pose in the sequence, $\hat{x}_{a+1}^{j\to k}$, which with a consistent map and poses, enables viewpoint selection and data association before using any information from appearance. The pose estimate $\hat{x}_{a+1}^{j\to k}$ can prespecify which of the landmarks L^k project onto $\mathcal{I}_{a+1}^{j}$ (similar to the depth map projection of LSD-SLAM [40]), which enables viewpoint selection (Sec. 7.1.1) without the use of image feature descriptors. Thus, viewpoint selection is appearance–invariant given a consistent map and poses. Occlusions can affect, however, the accuracy of viewpoint selection without an additional heuristic to further limit the set of points that is considered 'visible'. A simple heuristic to add is a constraint on the camera pose.

The pose estimate also prespecifies where the landmarks L_b^k project onto $\mathcal{I}_{a+1}^{j}$ for the 2D coordinates $\hat{\mathcal{M}}^{k,b\to j,a+1}$, which can be used to anchor dense correspondence (Sec. 7.1.2) before using any information about the visual appearance of the scene. Note, the dense correspondence obtained using map point anchors may not be appearance–invariant. The estimate of $\hat{x}_{a+1}^{j\to k}$ is computed using the pose transform, $(x_a^j \ominus x_a^{j\to k})$, and the known change in pose, $(x_a^j \ominus x_{a+1}^j)$, as

$$\hat{x}_{a+1}^{j\to k} = x_a^{j\to k} \oplus (x_a^j \ominus x_{a+1}^j). \tag{11}$$

This equation is equivalent to that of Eq. 10.

Reprojection Flow is used during the ISLC search where the relative poses between two surveys are estimated. Image retrieval is replaced with the viewpoint selection of Sec. 7.1.1 and full image alignment with the map–anchored dense correspondence of Sec. 7.1.2. Because Reprojection Flow boosts image alignment, the search for ISLCs proceeds backwards and forwards from a new ISLC, sometimes reattempting image alignment where it previously failed without map anchors. The use of Reprojection Flow in a particular search direction is stopped after unsuccessfully aligning three image pairs in a row or after encountering an ISLC.

8.4 Multi–Session Optimization

The third step of survey processing consists in applying multi–session optimization to acquire consistent maps and trajectories for a set of surveys. The ISLCs between them are the constraints that may indicate how to align them (see Fig. 45). Because visual feature descriptors are not shared among surveys, some of the ISLCs are *temporal loop closures*, which connect surveys at the beginning and the ends of the chain of surveys and may keep a long chain of surveys from drifting apart.

The constraints of the multi–session optimization can be represented using one large factor graph of multiple surveys and ISLCs, but the optimization is over subgraphs due to the need for scalability and robustness. Bundle adjustment applied to the full graph may otherwise become intractable in peak memory and optimization runtime as the number of surveys is increased [126, 113]. The full graph can be, fortunately, easily partitioned into subgraphs by replacing each ISLC with a pose prior (see Fig. 45). Subgraphs are thus optimized in parallel over several iterations. At the end of each iteration, the ISLC pose priors are updated using the result from the previous iteration. Compared to an optimization over the full graph, subgraph optimization can be lightweight, fast, and accurate [126].

The high likelihood of noisy ISLCs and the often weak constraints between poses

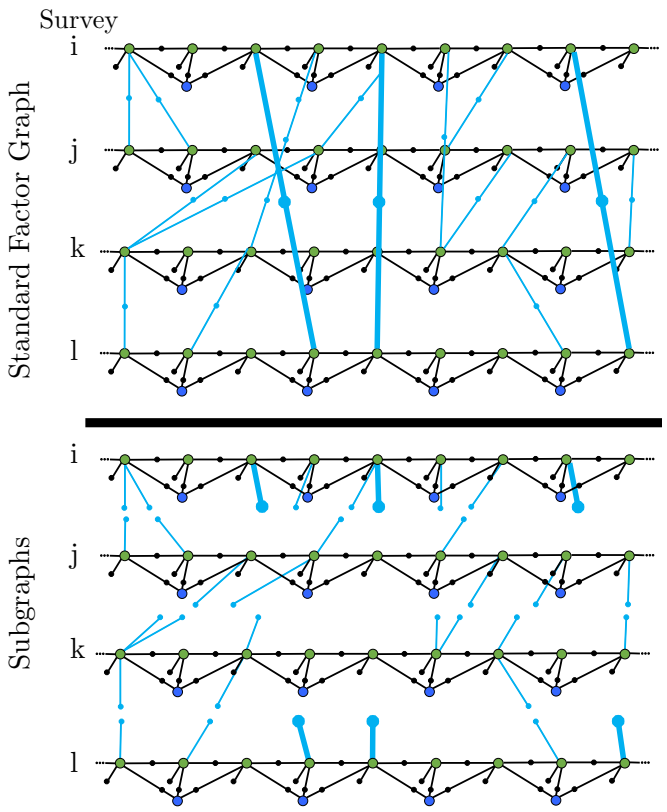

Figure 45: An example factor graph of the multi–session optimization and its conversion into subgraphs. The graph for each survey is nearly identical to that from single–session SLAM, in Fig. 16. However, instead of using velocity variables and a constant velocity assumption to constrain changes in camera poses, the changes in poses computed in Sec. 4 are used for that constraint. Blue lines represent loop closures between surveys. Thick blue lines delineate temporal loop closures, which are demarcated to bring attention to the fact that they may keep a long chain of surveys from drifting apart. Smart factors are used, but they are omitted in this visualization.

within each survey lead to the expectation maximization implementation of subgraph optimization. Expectation maximization is used to filter poor ISLCs over multiple iterations. An optimization that includes inaccurate ISLCs would otherwise pull surveys into an inaccurate alignment. Between the multiple iterations of the parallel bundle adjustment of subgraphs, before the ISLC pose priors are recomputed, the error of each ISLC is calculated to find outliers. Outlier ISLCs are deactivated for the next iteration.

The multi–session optimization is defined in Algorithm 1. For a number of surveys,

Algorithm 1 Subgraph multi–session optimization. The distance over the poses in line 9 are only over the position, not both the position and the orientation.

Input X^j, L^j, M^j, H^j for $j \in 1..n_\Pi$
Output $\mathcal{X}^j$, $\mathcal{L}^j$ for $j \in 1..n_\Pi$

1: $\{\mathcal{X}^j, \mathcal{L}^j\} \leftarrow \{X^j, L^j\}$ `for` $j \in 1..n_\Pi$
2: enum $state\{ALL = 0, FILTERED, DONE\}$
3: **for** $s = state::ALL; s \mathrel{!}= state::DONE; ++s$ **do**
4: **while** $\Delta\mathcal{C} > 0.01$ **do**
5: $H^j \leftarrow$`UpdateISLCs(`$H^j, \{\mathcal{X}^j, \mathcal{L}^j\}_{j=1}^{n_\Pi}$`,s)`
6: **for** $j \in 1..n_\Pi$ **do** ▷ in parallel
7: $G^j \leftarrow$`ConstructGraph(`X^j, M^j, H^j`)`
8: $\{\hat{\mathcal{X}}^j, \mathcal{L}^j\} \leftarrow$`BundleAdjustment(`$G^j$`)`
9: $c^j \leftarrow$ `Median(`$\{\| \hat{x}_t^j - x_t^j \|_2\}_{t=1}^{n_j}$`)`
10: $x_t^j \leftarrow 0.9 \times \hat{x}_t^j + 0.1 \times x_t^j$ `for` $t \in 1..n_j$
11: $\mathcal{C} \leftarrow \frac{1}{n_\Pi} \sum c^j$
12: **return** $\{\mathcal{X}^j, \mathcal{L}^j\}_{j=1}^{n_\Pi}$

n_Π, each one, j, with optimized trajectories, X^j, and landmarks, L^j, measurements of landmarks, M^j, and inter–session loop–closures, H^j, an iterative bundle adjustment is applied to recover the multi–session–optimized trajectories, $\mathcal{X}^j$, and maps, $\mathcal{L}^j$. Two stages of optimization are performed (referenced by *state*), which include a series of optimizations with every ISLC, followed by an expectation maximization series in which the inconsistent ISLCs are removed. Each series is implemented in multiple iterations (lines 4–11) with a weighted update (line 10) to gradually pull each survey into agreement with one another (a nonweighted update is susceptible to a nonconverging oscillation in some cases). Each survey is optimized in parallel (lines 6–10), with graph construction (line 7), bundle adjustment using the Levenberg–Marquardt algorithm (line 8), a measure of convergence (line 9), and a weighted update (line 10) applied separately to each survey. The optimization is considered converged when the median change in position, c^j, averaged over all the surveys, $\mathcal{C}$, has changed by less than 0.01 m (line 4).

Inconsistent inter–session loop closures (line 5) are filtered in the second stage to help boost map consistency. Incorrect ISLCs are identified using reprojection error.

For an ISLC between pose x_a^j and x_b^k, four different tests for outsize reprojection error are applied, which involve the 3D-to-2D point sets: 1) $(\mathcal{L}_a^j, \mathcal{M}^{j,a})$; 2) $(\mathcal{L}_b^k, \mathcal{M}^{k,b})$, 3) $(L_a^j, \mathcal{M}^{j,a \to k,b})$; and 4) $(L_b^k, \mathcal{M}^{k,b \to j,a})$. A threshold is computed using the reprojection error, r_a^j, which is measured using $(L_a^j, \mathcal{M}^{j,a})$. If any of the four tests of reprojection error exceed $3 \times max(r_a^j, \frac{1}{n_\Pi} \sum_j \frac{1}{n_j} \sum_a r_a^j)$, the ISLC is marked as an outlier and goes unused until some later update changes it back.

8.5 Experiments

The experiments evaluate the new methods on the Symphony Lake Dataset (Sec. 3). The framework is first shown apply to a dataset of that size and complexity (Sec. 8.5.1). It finds abundant data association across its images, and can optimize all the maps and trajectories in tractable time. The map was next used with Reprojection Flow to align random image pairs across different time intervals. The results were compared to related approaches to show by how much the map helped image alignment (Sec. 8.5.2). The comparison goes one step further to show that the well–aligned images from the methods of this chapter are more often superior (Sec. 8.5.3). Image alignment quality was then divided by scene, which showed that many well–aligned images could be expected in many time–lapses (Sec. 8.5.4). From there, 100 time–lapses of random scenes were produced, of which several had a large number of well–aligned images (Sec. 8.5.5). With promising results, the pose error of misaligned image pairs was evaluated, which showed that improving map consistency in future work could lead to even better results (Sec. 8.5.6).

8.5.1 Aligning One Year of Surveys

The framework was first applied to each year of surveys of the Symphony Lake Dataset to characterize its runtime and data association performance. The framework was applied four times for the four years of surveys between 2014 and 2017. For one survey from the dataset, a run of single–session SLAM had peak memory usage of nearly

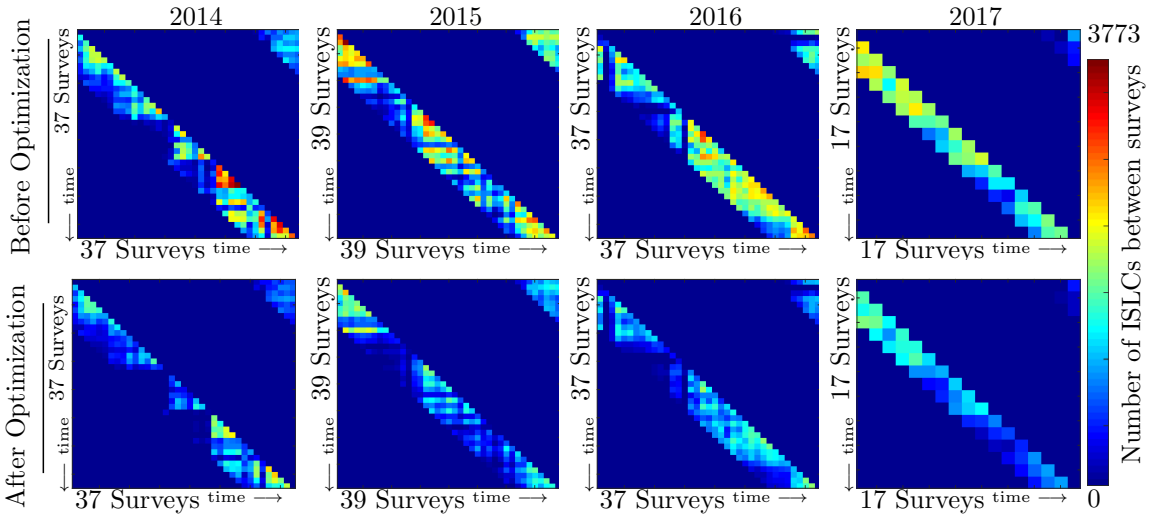

Figure 46: Inter–session loop closure connectivity for each year of the Symphony Lake Dataset **top)** before and **bottom)** after optimization. Each grid cell represents the number of ISLCs between two surveys. A figure has more grid cells if that year had more surveys. The grid cells for 2017 are larger because there were fewer surveys in that year. The ISLC search was also limited to three, rather than eight, surveys because they were captured less frequently. The nonzero cells in the top-right of each grid account for the ISLC connectivity across the time between the beginning and the end of each year.

16GB and completed in about two minutes (on a 2.4GHz machine). The average reprojection error of an optimized map and camera trajectory was approx. 3.5 pixels [59], which indicated that each map–trajectory tuple was individually consistent.

The data association pipeline of Sec. 8.2 was effective in providing a large number of inter–session loop closures between surveys of a natural environment, as shown in the top row of Fig. 46. For the Symphony Lake Dataset, the image alignment pipeline was only run between surveys within three months of each other, which are the ones that typically had appearance–based alignments. The few ISLCs that could have been obtained beyond that was not worth the extra runtime. Because surveys were captured roughly bi-weekly, ISLC search was run from each survey to each of its eight previous surveys. The search was similarly applied to pairs of surveys between the beginning and the end of the year, which created a temporal loop–closure. For the set of surveys from 2014, for example, Sec. 8.2 was applied to a total of $37 \times 8 = 296$

pairs, which resulted in 332,441 ISLCs. The runtime on an average pair of surveys took approx. 5-7 hours (mostly consumed by SIFT Flow). A cluster of 20 nodes was used to collect the constraints for each year of surveys in approx. one week. The use of Reprojection Flow within ISLC search added approx. 1.3× more ISLCs.

A large number of inter–session loop closures were also retained after multi–session optimization, as shown in the bottom row of Fig. 46, which may represent a consistent set. Approximately 58% of the ISLCs were retained. After 6-10 iterations with all the ISLCs, each of the sets took five more iterations to converge again with filtering. Each survey was optimized in 30-45 seconds, with each iteration of multi–session optimization taking double that for 37 surveys on a machine with 32 threads (the runtime of optimization was in proportion to the number of surveys and the number of machine threads). The update step of the filtering stage (line 5 of Alg. 1) added approx. 1 minute to each iteration. The total optimization runtime was approx. 25 minutes for a set of 37 surveys. For comparison, a single iteration of bundle adjustment over the standard, full graph (shown in the top half of Fig. 45) took longer than 24 hours so was terminated.

The patterns of connectivity varied for different pairs of surveys, but were similar before and after optimization. Connectivity decreased between pairs away from the diagonal, consistent with the increased amount of time between surveys. It also dropped out between surveys with large differences in lake levels, notably to a group of surveys in 2014 and to two surveys in 2016. Image pairs between those surveys had the most variation in appearance. The matching pattern of connectivity after optimization indicates that the multi-session optimization result had a similar goodness-of-fit to the constraints among all the surveys. The evaluation does not indicate, however, how consistent the map is.

8.5.2 Image Alignment Quality

The map consistency is measured using the image alignment quality for image pairs of random scenes between random surveys. An indirect approach is used because the dataset lacks a ground–truth localization result, as could be provided by an RTK GPS system. In this evaluation, images are aligned and hand–labeled to measure how consistent the multi–session optimization result is. For the comparison, 1000 random image pairs of the same scenes were selected, aligned, flickered back and forth in a display, and then manually labeled well–aligned or not for four different methods:

SF* SIFT Flow with image alignment constraints

RF* Reprojection Flow with image alignment constraints

RF Reprojection Flow without constraints

ICP-H an image alignment approach based on the use of ICP (on 2D LiDAR data), a homography, and multi–session optimization from [135], to which SIFT Flow was added, which can make the alignment more precise.

Pradalier and Pomerleau [135] produced a consistent map and trajectories of the Symphony Lake Dataset by applying an ICP algorithm to the 2D laser scan data of each survey, which produced a result they used to facilitate image alignment. They first applied ICP to the laser scan data to get pose transforms between images from different surveys. The sequence of 2D transforms were added to a factor graph of keyframes, one every 20 m and 1 minute degree, which was optimized for all the surveys in a year. After multi–session optimization, they showed that an image pair of the same scene could be nearly aligned by applying a homography to parallelize the image planes. Indeed, a homography also removes changes in scale that can affect SIFT feature matching. Thus, for the comparisons in this chapter, SIFT Flow with a small hypothesis space was added (see the bottom row of Fig. 47), which can limit

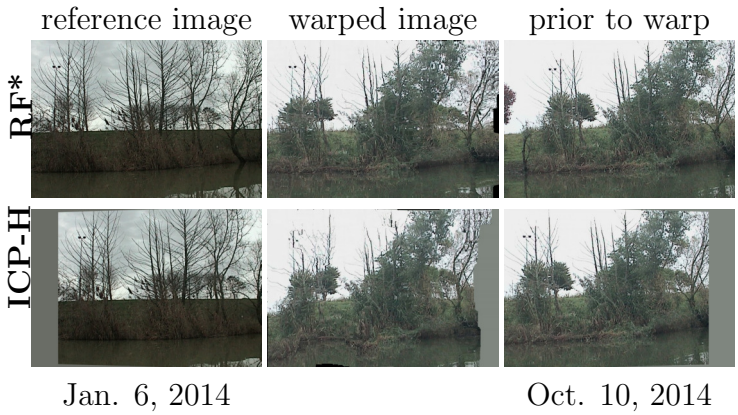

Figure 47: Example comparison of image alignment using RF* vs. using ICP-H (The method of [135] to which SIFT Flow was added, which can make the alignment more precise). Whereas RF* uses the reprojection of map points to set the hypothesis space, in the latter approach, a homography is applied to parallelize the image planes. The ICP-H image pair may be nearly aligned. To this ICP-H pair, however, alignment using SIFT Flow was added. In this figure, only the image pair aligned using RF* is well–aligned. The ICP-H approach set the hypothesis space to the wrong regions of the two images.

perceptual aliasing, can keep the alignment quality higher on average, and can make for a fairer comparison.

The four methods are distinguishable in the number of high–quality alignments they produced, as shown in Fig. 48. SIFT Flow produced significantly fewer well–aligned images, which shows that using a map to guide image alignment was, for these cases, significantly better than not. The best method at every time interval was Reprojection Flow, which relied most on the map to guide image alignment. The dip in the alignment quality towards six months indicated that the variation in appearance had an effect on all four methods.

8.5.3 Comparing Reprojection Flow to the ICP-Homography Approach

The number well–aligned images of [135] showed that its performance was in many cases close to Reprojection Flow, which motivated a direct comparison of the aligned images to better gauge any difference in alignment quality. The aligned image pair of both methods were placed side-by-side in a flickering display. The result that better

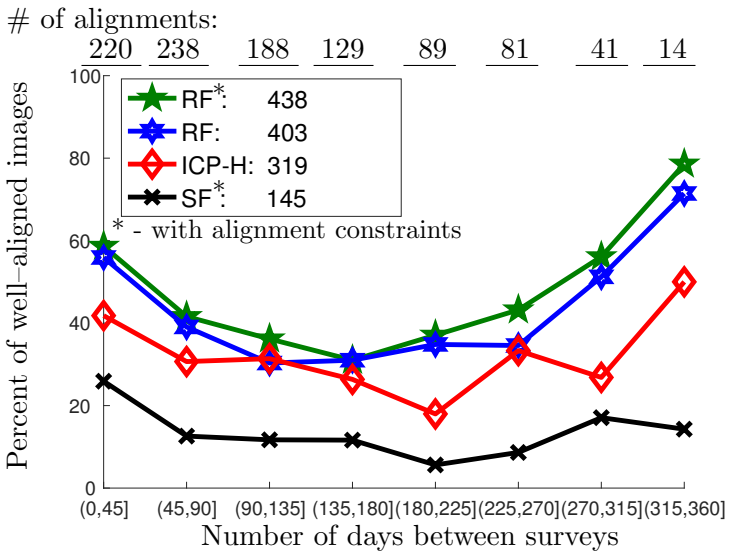

Figure 48: Comparison of alignment quality over time shown as the percent of well–aligned images per time interval. A single alignment was of two images of the same scene taken from two different surveys. Each method aligned the same set of 1000 random image pairs, generated from the 2014 surveys from the Symphony Lake Dataset. The top row shows the number of alignments in each time interval. The y-axis plots the percent of those well–aligned images.

aligned the scene contents was manually identified. Otherwise, if neither was better than the other, the pair was labeled comparable. The process was repeated for all 1000 image pairs of Sec. 8.5.2.

The result in Fig. 49 shows that the two methods produced comparable image alignments in about half the cases. For the rest of the image pairs, Reprojection Flow produced better alignments twice as often as ICP-H, a trend unaffected by the change in time scales in a year. Most of the differences in image alignment quality appeared to derive from differences in map consistency. Where the maps were incorrect, the images were setup for an incorrect alignment, as shown in e.g. the bottom row of Fig. 47. Many of the comparisons were often between image pairs where neither aligned well, but some parts of one image pair aligned well. That made the trend in this figure different from that of Fig. 48.

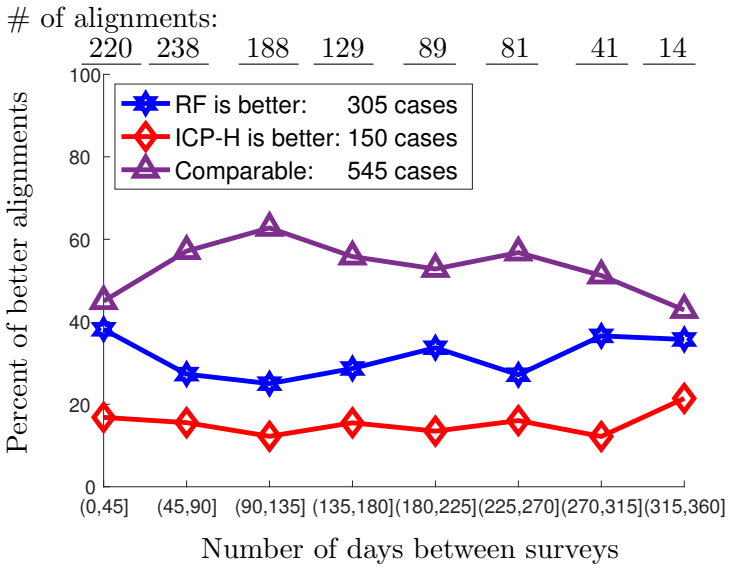

Figure 49: Grading Reprojection Flow to ICP-H by comparing 1000 random image pairs that were aligned with both methods. Reprojection Flow was applied without alignment constraints for this comparison, which kept its function closer to that of ICP-H. The comparison divides the 1000 image pairs into eight intervals of time between surveys, in increments of 45 days, to show that the trend was unaffected by the variation in appearance.

8.5.4 Image Alignment Quality By Scene

Next, the image alignment accuracy was plotted by scene to determine by how much different scenes affected alignment performance, and where complete time–lapses (with an image from every survey) could be possible. The average image alignment quality of 43.8% suggested that complete time–lapses could not be produced unless the images aligned better at different scenes, which was likely. A cover set of the environment was identified for one survey (the cover set was acquired by applying the method of [59] to the June 25, 2014 survey) and then each of the 1000 image pairs was added to the nearest scene (the position where the reference image had the min L_2 distance). If the scene had at least four image pairs, then the percent of well–aligned image pairs was plotted.

Image alignment quality varied substantially by scene, as shown on the left side of Fig. 50. Some scenes along the shoreline had many well–aligned images whereas

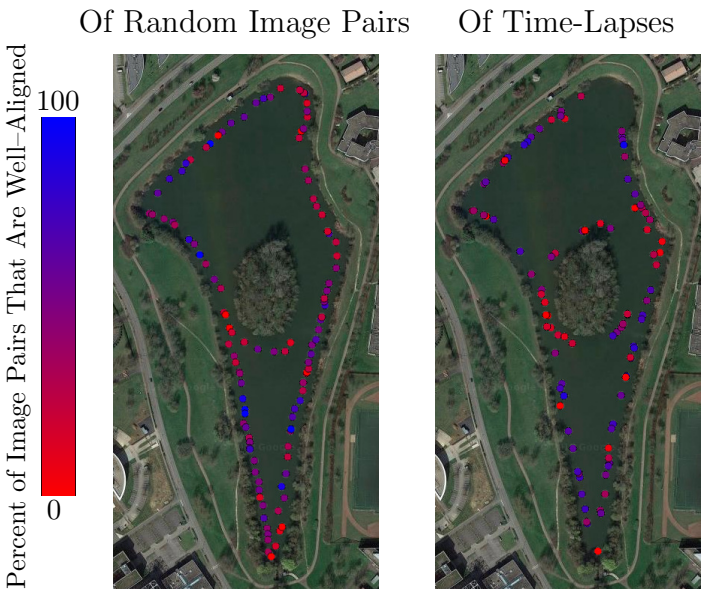

Figure 50: Alignment quality around Lake Symphony for the **left)** image pairs of Sec. 8.5.2 and the **right)** time–lapses of Sec. 8.5.5. The similarity of the results of the two sets indicates that Reprojection Flow may be robust to difficult variation in appearance at some locations. The satellite view is from Google Maps.

other scenes had none. The scenes with the most well–aligned images were along the straights. Fewer well–aligned images were produced along curves in the path. Thus at certain locations the map–anchored approach to dense correspondence showed robustness to difficult variation in appearance. Yet, the number of well–aligned image pairs did not yet demonstrate that the method was robust to a full year of variation in appearance, or if those were locations where a large number of image pairs were from sessions captured around the same time. The next question was whether this result held true across complete time–lapses with a year of variation in appearance.

8.5.5 Producing Time-Lapses

With a high likelihood of more complete time–lapses at some scenes in the environment, the next step was to create them. To create a time–lapse, a reference image was randomly chosen from a random survey, and then an image of the same scene from every other survey was selected and aligned using Reprojection Flow with constraints.

Figure 51: Timelapse of one scene of Symphony Lake from 32 surveys captured between 2014 Jan. 6 and 2014 Dec. 22. The images were selected and aligned to the reference image using Reprojection Flow.

The quality of the time–lapse was manually labeled. First the reference image and each image of the time–lapse were flickered to keep only well–aligned image pairs. Then the time–lapse was repeatedly scrolled through to keep only the images that added to it (also well–aligned). Two examples are shown in Figs. 51 and 52. The process was repeated 100 times.

The results show that the quality of the time–lapses was consistent with the image alignment quality of Sec. 8.5.2; although the time–lapse quality varied, some locations aligned particularly well. The quantitative time–lapse quality is shown in Fig. 53 and shown by place in the right of Fig. 50. Approximately a third of the time–lapses had about two thirds or more of well–aligned images. These time–lapses typically spanned all four seasons. The misaligned image pairs did not consistently have significantly more variation in appearance, and were not always from consecutive surveys. Instead, the reprojected map points appeared to mismatch the correct alignment (see Fig. 54 **top)**).

Some effects of aligning a set of images into a time-lapse include the lack of variation in viewpoint and the addition of noise in the result. Before applying image

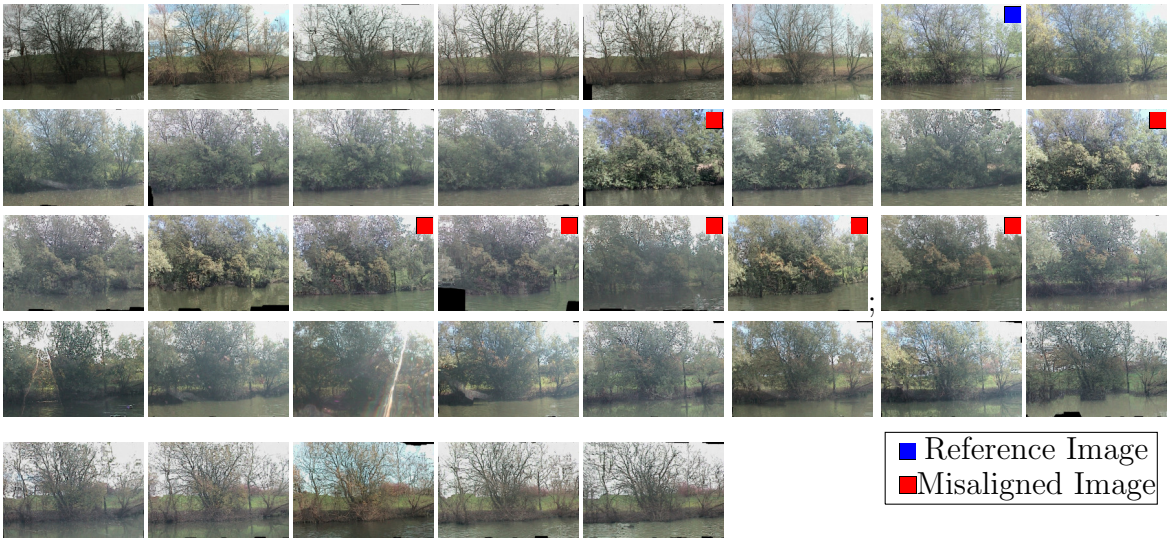

Figure 52: Timelapse of one scene of Symphony Lake from 37 surveys captured between 2014 Jan. 6 and 2014 Dec. 22. The images were selected and aligned to the reference image using Reprojection Flow.

alignment, a set of images of a scene was a time–lapse whose variation in viewpoint sometimes detracted from the collection. After, the noise added due to the alignment process sometimes detracted from it (see Fig. 54 **bottom)**). Having accurate maps and very similar viewpoints helped minimize that noise to create visually smooth transitions. Noise often was, however, a side effect of image alignment.

8.5.6 Pose Error of Misaligned Image Pairs

Because any pose error could cause errors in the locations of reprojected map points and could lead to misaligned images, the next evaluation measured the magnitude of the pose error for misaligned image pairs taken from the time–lapse set. A misaligned image was selected for the evaluation if it appeared to share several strong features with the reference image, which simplified the labeling task. A map point in the reference image was selected and the corresponding point in the selected image was hand–labeled. After hand–labeling at least 15 correspondences, a one-way localization was performed, similar to that described in Sec. 8.2.3.2. The localized pose was used as the ground truth if the map points from the reference image projected onto their locations in the selected image. If the projected map points were incorrect, the set

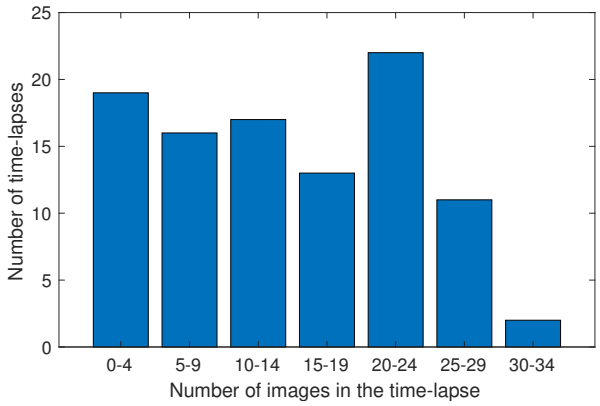

Figure 53: Success of 100 random time-lapses measured as how many images each one consisted of. About a third of them had 66% or more well–aligned images. Most of the time-lapses were created from approximately 33 image alignments. Although Symphony Lake Dataset had 37 surveys from 2014, typically only about 33 captured the same scene.

of hand–labeled correspondences was refined until they did or were discarded. The process was repeated for 100 misaligned image pairs.

Figure 55 shows that the pose error was nonnegligible for almost all of the misaligned images. The poses of misaligned images had a median translation error of 1.06 m and a median orientation error of 3.15 degrees. Pose error this high caused reprojected map points to be far off from their correct locations. Even for the pose with the least error of 12cm and 1/2 a degree, error in the reprojected map points was visible as slight misalignment of more distant scene contents. For that image, however, the variation in viewpoint was also a primary cause of misalignment. Although a foreground object was aligned well, the background behind it was pulled out of alignment. Images from more similar viewpoints with accurately projected map points aligned best.

8.6 Conclusion

This chapter confirms that map point correspondence priors and geometric constraints within a dense correspondence image alignment optimization could be used to achieve

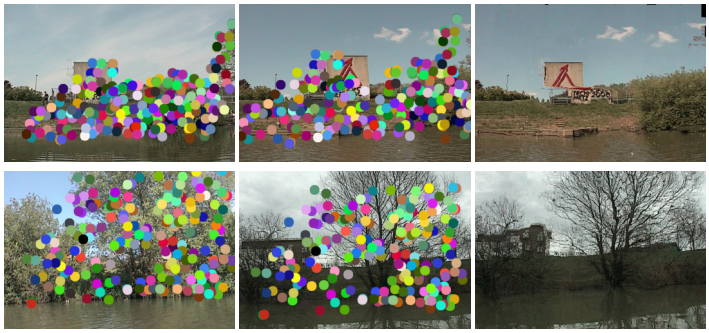

Figure 54: Noise in aligned images. **(top)** Map points from two surveys are projected onto the reference image (left) and the image to be aligned (middle). Their inconsistency caused the error of the aligned image (right), which otherwise had a strong appearance–based correspondence. **(bottom)** The alignment process added noise to the tree structure in the well–aligned image (right), even though the map point priors were consistent.

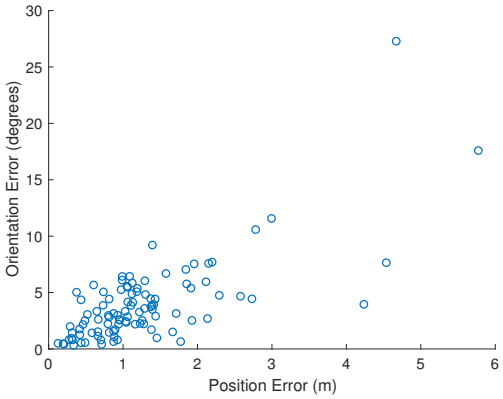

Figure 55: Pose error for 100 misaligned image pairs. The median error of the points here is 1.06 m and 3.15 degrees.

data association across the year–long variation in appearance of a natural environment. A substantial effort at creating and relying on geometric information and data association constraints was key to acquiring accurate loop closures. Reprojection Flow was then key for bringing more difficult image pairs into alignment. Using a multi–session optimization algorithm that filtered outlier loop closures as it divided–and–conquered the problem helped lead to a highly consistent map for a year of surveys. Although a long time–lapse at every location along the shore was not produced, the results show promise on how to obtain one from a consistent map of a

natural environment and the dense correspondences of its images.

The next chapter is the conclusion, which provides a conclusion in the larger context of this thesis, describes limitations of this approach, and suggests directions for future work.

CHAPTER IX

CONCLUSION

9.1 Conclusion

A foundation for data association across seasons and viewpoints is the spatial information in individual scenes and an environment map. This conclusion is supported in two key ways.

1) Whole images captured the manifold structure of a scene, which enabled dense correspondence to provide greater robustness to variation in appearance between surveys than two other approaches for data association. With robust inter–session loop closure search built upon dense correspondence, it was possible to identify images of the same scenes, acquire many accurate loop closures, and recover a single map of multiple surveys from different seasons.

2) Reprojected map points guided dense correspondence across seasons, which lead to year–long time–lapses at many different scenes of a natural environment. Through the Reprojection Flow algorithm, the co-visibility of reprojected map points enabled appearance–invariant viewpoint selection, which identified images closer to the same scenes than a pose–distance heuristic. Reprojected map points also provided priors for robust dense correspondence, which anchored the data association search even in the midst of significant variation in appearance. Without Reprojection Flow, dense correspondence power was limited to approximately three months of variation in appearance.

These results were made possible by the new Symphony Lake Dataset, a public dataset of a large–scale natural environment captured by a seeing, mobile machine in

multiple deployments over a long period of time. No other public dataset has as many surveys, which accumulate as much variation in appearance and viewpoint, as this dataset. This dataset is also unique because it captured surveys of an environment; the camera pointed sideways to the robot's direction of motion. As Chapters 4, 5, and 6 showed, achieving accurate visual data association beyond three months of surveys may represent new feats in robustness, yet this dissertation achieved data association over all the surveys in a year.

The foundation provided by position–based dense correspondence was successful because it cut through multiple factors of variation in appearance of a natural environment for robust visual data association. Dense correspondence was effective in being guided by map point priors, but it could also combine that with appearance in an optimization-based data association. This combination was able to produce robust visual data association that would have otherwise taken an assortment of methods to account for each factor of variation separately (e.e., illumination, noise, raindrops, weather condition), or training data to factor out the general variation (e.g., using PCA or a neural network). Whereas a position–based approach using dense correspondence may largely be unaffected by new factors of variation (as long as visual SLAM has accurately captured the environment), other approaches may fail if there are any factors of variation that are unaccounted for (either in the noise–reducing methods or in the training data). This supports the idea that a position–based dense correspondence may have properties of appearance–invariance.

The foundation provided by map point priors was successful because it defined correspondence at times when the appearance lost matching power. Landmarks captured the environment through long feature tracks regularly spread out across long image sequences, leading to accurate 3D maps of the whole environment. Before Reprojection Flow, this information was not native to dense correspondence. When

they were integrated, reprojected landmarks helped bridged some gaps in correspondence where the map point priors were accurate and the viewpoints were similar. It also reduced perceptual aliasing among highly self–similar scenes. In other words, reprojected map points were a powerful supplement to appearance–based dense correspondence, which enabled new feats of accurate visual data association in a natural environment.

The successful production of multiple time–lapses at many different locations along a natural environment is evidence for the power of environment structure for visual data association. Time–lapses aligned images of the same scenes from multiple surveys of a natural environment across all the seasons in a year. The results resembled (with image alignment noise) image sequences captured from multiple stationary cameras. At a time when frequent advancements are made towards robust visual data association, the spatial information in a map may be able to close the distance where hard cases have persisted between observations.

9.2 Limitations

The advancements of this dissertation are limited in a few ways, which should be considered in new applications that seek to deploy or build upon this work. The primary limitations are the conclusions drawn from one dataset (Sec. 9.2.1), in the dense correspondence of images from different viewpoints (Sec. 9.2.2), in using a map for data association (Sec. 9.2.3), and in the assumption of Reprojection Flow that it builds on a consistent map (Sec. 9.2.4).

9.2.1 Conclusions From One Dataset

The conclusions were drawn from results on a single dataset, which is one particular type of natural environment, captured in one type of way. The USV was 10 m from the shore, its range of motion was limited to a 2D plane, it moved at a steady, slow speed, the camera resolution was low, and it lacked the ground–truth trajectory of

each run. Significantly more variation in viewpoint with less image overlap would have been likely if, for example, the same environment was surveyed from a drone. A different natural environment with different scene layouts could have been more difficult for image alignment and visual SLAM to capture the scene structure. Yet, the results succeeded in showing the difficulty of data association across seasons in this dataset, and that the methods from this dissertation addressed that challenged better than several other methods.

9.2.2 Limitations of Dense Correspondence

Some of the errors in dense correspondence were due to limitations of the optimization, which tries to align every pixel in scenes with objects at different depths, even if they are captured from different viewpoints. Images from different surveys may not always have a complete dense correspondence. As some distant objects become occluded and others spontaneously appear from behind foreground objects, the mapping of a 2D image onto another lacks the descriptive power to capture and represent the occlusion. To exacerbate the limitation, the smoothness assumption of the dense correspondence optimization may wrongly ensure the mapping is contiguous through the change.

This limitation directly affects the applicability of the dense correspondence method. If it has not demonstrated a lower degree of robustness to variation in viewpoint, the images may have captured a scene that is primarily at one particular depth. It may be more difficult to align images captured in e.g., dense forests, where there are many thin objects at largely varying depths, with high perceptual aliasing. The closer the images from different surveys are to the same viewpoints, the more of the same content will be captured in the images, and the more accurate the alignment may likely be. Otherwise, while large parts of the foreground may be well aligned, more distant objects may have been spuriously associated.

9.2.3 Limitations of Using a Map To Guide Dense Correspondence

Using map point priors to guide dense correspondence is limited to the range of scenarios for which a consistent map of multiple surveys can be acquired. Acquiring a year of surveys of a natural environment before gaining the boost from map point priors is probably unrealistic. Acquiring more frequent surveys (e.g., bi-weekly or monthly, as opposed to seasonally) in order to facilitate dense correspondence, rather than to facilitate a surveying need, may also be cumbersome. Until these limitations are addressed, a better way to use map point priors may be to help fill in gaps of appearance–based dense correspondence between consecutive surveys.

With enough surveys to get a consistent map, a second limitation is in the computation time required to get all the loop closures. A search for loop closures between two surveys is expectable, as new observations are typically related to a prior set in this way. But because each survey has its own, independent map, there is a potential need to run the search between more pairs of surveys (e.g., in case two surveys have fewer loop closures), which can quickly become cumbersome. In this dissertation the search was run between each survey and its eight prior surveys to ensure there were enough constraints to build a consistent map. The computation time and the robustness of dense correspondence was the primary bottleneck.

9.2.4 Limitations of Reprojection Flow

Any benefit derived from Reprojection Flow is due to the consistency of the maps and the poses. Any inconsistency could lead to inaccurate viewpoint selection (a pose–based heuristic would be more accurate) and inaccurate dense correspondence. Reprojection Flow has a lot of power to define where data association should occur, either when used between a new survey and a prior map, or between multiple consistent maps. The experiments of Chapter 8 showed that time–lapses were inaccurate where the maps and poses were inconsistent. Thus, Reprojection Flow is only

beneficial in applications where a consistent map and poses can be acquired.

Chapter 8 used Reprojection Flow during the loop closure acquisition search because it sped up the loop closure search time, often made it more robust to variation in appearance, and it resulted in 1.3 times more loop closures, but its use during the search risked locking the search into a series of inaccurate loop closures. The appearance may become, in some cases, too nondescriminative to pull dense correspondence away from the map point priors (e.g., a transition to all gray images after reaching a verified localization). This is different from the case of images from different seasons, whose SIFT images may mismatch, but which may be highly discriminative. Discriminative appearance features counteract inaccurate map point priors. In cases where a series of inaccurate map point priors have resulted in a series of inaccurate loop closures, their inconsistency with the larger set of loop closures among all the sessions could lead to their removal during multi-session optimization.

9.3 Future Work

While many results were achieved in support of the hypotheses, a number of areas of improvement were identified and left for future work. The improvements extend the algorithmic foundation to make it more robust and practicable.

A seeing, mobile machine often has more sensors than a basic camera (e.g., a multi-spectral camera; a LiDAR), which often provide equally powerful information that can be integrated with the methods of this dissertation for improved dense correspondence. A laser range finder, for example, provides a source of information with which multiple surveys may be made consistent [135]. It could thus help with establishing dense correspondence over time between surveys with large degrees of variation in appearance. More sensors may also help to establish loop closures across periods of time beyond e.g., a year, when the similarity in the visual appearance has started to fade.

Image alignment from dense correspondence should be formulated to handle occlusions differently. Currently, dense correspondence is set to a reverse rather than to an onto mapping. A mapping 'onto' could leave holes in the image. Yet, leaving holes would be more accurate than blending that content with nearby content of the same image. A better alternative is to use inpainting to fill holes when the occluded region is visible in nearby images (as in e.g., [86]).

A prealignment step should be applied before dense correspondence in order to improve matching power between images with variation in scale and rotation. The SIFT descriptors computed for matching with SIFT Flow lacked the scale and the orientation invariance due to the lack of keypoint search. SIFT features were extracted at each pixel at the same scale and the same, upright orientation. Because of that, any variation in the scale or in the rotation leads to even more variation in the descriptors that are meant to be matched with one another, which makes matching more difficult. Because the maps and the poses are known a priori, however, two images may be pre–aligned using a homography to invert the variation. Indeed, this pre-alignment was implemented in related work [135, 34]. It could add a similar benefit to future versions of the work from this dissertation.

The lack of loop closures between some surveys indicates that appearance–based correspondence is in need of more data association power. SIFT features are the foundation for SIFT Flow, which have been outclassed by recent advancements towards learned feature descriptors. Superpoint [31] or FCSS [84] features could replace SIFT descriptors (especially if trained on data from natural environments [102]), while retaining the original optimization of SIFT Flow (belief propagation over an MRF). Newer optimization frameworks for dense correspondence, which estimate local affine transformation fields between images, have demonstrated improved matching power over that of SIFT Flow even if SIFT Flow has a new feature descriptor [85]. Map point priors may also be integrated into a new dense correspondence framework [33].

Collectively, these advancements promise a greater set of more accurate loop closures between surveys with more variation in appearance.

A number of other changes may be made to the framework for improved dense correspondence among multiple sessions, which are listed here. Superpoint [31] has been shown to outperform KLT feature tracking; it may provide more points, each with descriptive power between surveys, for a more descriptive map. Additional map point visibility heuristics (e.g., a pose heuristic, or a visible tracked point) could improve the viewpoint selection of Reprojection Flow to better define the visible set of map points in each image. The factor graph of Multi–session optimization could include a reference frame between each pair of surveys, which would help to eliminate outlier loop closures.

Finally, perhaps the simplest, and most impactful improvement for future work would be to form a larger ground–truth dataset of localized poses between surveys of the Symphony Lake Dataset. The lack of accurate ground truth poses was somewhat inhibitive to progress because method evaluation took much longer. (Hand-labeled image alignment was time–consuming and prone to a small bias. New algorithms were evaluated more slowly.) There are currently 100 known ground–truth localizations between surveys, which were obtained by hand–labeling correspondences and then solving for the localized poses. A set of 1000 or more may be more appropriate for targeted evaluations, given that there are 130 surveys. Forming the set simply requires the time for hand–labeling more correspondences between images. Once a set is acquired, other researchers may also more easily benefit from an evaluation on the Symphony Lake Dataset.

APPENDIX A

PARAMETER VALUES

A number of parameters were defined and tuned in the making of this dissertation. Parameters for the final framework, in Chapter 8, are summarized in Table 5. The framework has many parameters because it has many different steps. SIFT Flow and RANSAC were the two off-the-shelf methods. Factor graph optimization was implemented using GTSAM, whose noise models are left out here. Each step was typically tested and tuned individually and then their parameter values were left as-is after integration.

Parameter values were found that applied well within the Symphony Lake Dataset of Chapter 3. Some parameters were made adaptive if a particular setting was insufficient. For example, the inlier/outlier threshold at the end of Sec. 8.4 on multi–session optimization was initially a single value of six pixels, but that was inconsistent with the varied reprojection error after single–session SLAM. Other parameter values may change when this framework is applied to different datasets. In that case, the pixel value limits could be made proportional to the image resolution. However, most may have broader applicability.

Although the alignment constraints have more parameters than the other steps of the framework, the formulation became more general. Image alignment verification in Chapter 8 replaced the use, in Chapter 7, of an alignment energy threshold (of 1120000) and an alignment consistency threshold (of 95%) to distinguish well–aligned images. Those thresholds applied well to a 100m section of shore that was initially evaluated against. For larger stretches they were ineffective because those values do

not robustly correspond to well–aligned images. The use of 19 iterations of two-cycle consistency was, however, retained because testing showed that the consistency typically converged (but not necessarily to 95% or more pixels) before that or not at all. The 95% threshold was set because the first layer of image alignment is an approximation, where a 95% consistency was, for images from the Symphony Lake Dataset, correct enough to proceed with the alignment of larger resolution layers.

More tuning was done to find the 40% threshold than to tune the $(3, 3)$ pixel shift that was used to verify an alignment. The shift was used to address perceptual aliasing. Aliasing occurred in data association due to the reflective lake on the bottom of images and due to the similar shore contents to the sides of the images. Shifting the images was a way to identify whether perceptual aliasing occurred. The best threshold was found by inspecting the results.

Table 5: Summary of parameters for the different parts of the final framework. Factor graph weights are omitted.

Sec. 6.2.1: SIFT Flow	
$\alpha = 255$,	alignment energy function parameters
$d = 10200$	
$\nu = 0.255$	
h	= hypothesis space size down the image pyramid
11,5,3,1	
100	iterations of message passing
Sec. 7.2.2: Alignment constraints	
(3,3) pixels	image translation applied for alignment verification
40%	min proportion of matching correspondences at which an image alignment is verified
at most 19	iterations of two–cycle consistency
95% $\leq$ 1 pixel	stopping criterion at which the forward and reverse flows are consistent
16	multiplier weight for the cycle consistency term
2.5	value for the epipolar constraint term
Sec. 7.2.3, 8.2.3.1, and 7.1.2: Fundamental matrix estimation	
3 pixels	RANSAC error threshold
0.999	probability the fundamental matrix is correct
Sec. 8.2.1: Image Retrieval	
5 m, 20 deg.	max pose distance to consider an image for alignment
3	max consecutive attempts of using RF during the search without success
Sec. 8.2.3.2: Localization	
15	number of correspondences used for localization in each iteration of RANSAC
100	iterations of RANSAC
6 pixels	max acceptable reprojection error of an inlier 3D-to-2D correspondence
Sec. 8.2.3.3: Bi-Directional Refinement	
at most 15	iterations of expectation maximization
40%	min proportion of inlier correspondences at which localization is successful
Sec. 8.2.3.4: Loop–closure verification	
25%	min image overlap required of two localized poses to verify them
6 pixels	max average reprojection error at which two loop–closures agree
Sec. 8.4: Multi–session optimization	
3	multiplier for the ISLC inlier/outlier threshold
0.01 m	convergence criterion as the change in the average median change in position
0.9	weight for the weighted update

REFERENCES

[1] Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W., "Robust map optimization using dynamic covariance scaling," in *IEEE International Conference on Robotics and Automation*, pp. 62–69, 2013.

[2] Amini, A., Rosman, G., Karaman, S., and Rus, D., "Variational End-to-End Navigation and Localization," in *IEEE International Conference on Robotics and Automation*, 2019.

[3] Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J., "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5297–5307, 2016.

[4] Arroyo, R., Alcantarilla, P. F., Bergasa, L. M., and Romera, E., "Towards life-long visual localization using an efficient matching of binary sequences from images," in *IEEE International Conference on Robotics and Automation*, pp. 6328–6335, 2015.

[5] Bargoti, S., Underwood, J. P., Nieto, J. I., and Sukkarieh, S., "A pipeline for trunk detection in trellis structured apple orchards," *Journal of Field Robotics*, vol. 32, no. 8, pp. 1075–1094, 2015.

[6] Beall, C. and Dellaert, F., "Appearance-based localization across seasons in a Metric Map," in *IEEE/RSJ IROS Workshop on Planning, Perception, and Navigation for Intelligent Vehicles*, 2014.

[7] Beery, S., Van Horn, G., MacAodha, O., and Perona, P., "The iwildcam 2018 challenge dataset," *arXiv preprint arXiv:1904.05986*, 2019.

[8] Benbihi, A., Geist, M., and Pradalier, C., "Elf: Embedded localisation of features in pre-trained cnn," in *IEEE International Conference on Computer Vision*, 2019.

[9] Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., Levine, S., and Vanhoucke, V., "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *IEEE International Conference on Robotics and Automation*, pp. 4243–4250, 2018.

[10] Bradski, G., "OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[11] Bradski, G. and Kaehler, A., *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[12] Bruce, J., Wawerla, J., and Vaughan, R., "The SFU mountain dataset: Semi-structured woodland trails under changing environmental conditions," in *IEEE ICRA Workshop on Visual Place Recognition in Changing Environments*, 2015.

[13] Bryson, M., Reid, A., Ramos, F., and Sukkarieh, S., "Airborne vision-based mapping and classification of large farmland environments," *Journal of Field Robotics*, vol. 27, no. 5, pp. 632–655, 2010.

[14] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[15] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., "Brief: Binary robust independent elementary features," *European Conference on Computer Vision*, pp. 778–792, 2010.

[16] Carlevaris-Bianco, N. and Eustice, R. M., "Generic factor-based node marginalization and edge sparsification for pose-graph slam," in *IEEE International Conference on Robotics and Automation*, pp. 5748–5755, 2013.

[17] Carlone, L., Dong, J., Fenu, S., Rains, G. C., and Dellaert, F., "Towards 4d crop analysis in precision agriculture: Estimating plant height and crown radius over time via expectation-maximization," in *IEEE ICRA Workshop on Robotics in Agriculture*, 2015.

[18] Carlone, L. and Calafiore, G. C., "Convex relaxations for pose graph optimization with outliers," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1160–1167, 2018.

[19] Carlone, L., Kira, Z., Beall, C., Indelman, V., and Dellaert, F., "Eliminating Conditionally Independent Sets in Factor Graphs: A Unifying Perspective based on Smart Factors," in *IEEE International Conference on Robotics and Automation*, 2014.

[20] Chahine, G. and Pradalier, C., "Survey of monocular slam algorithms in natural environments," in *IEEE Conference on Computer and Robot Vision*, pp. 345–352, 2018.

[21] Chen, Y.-C., Huang, P.-H., Yu, L.-Y., Huang, J.-B., Yang, M.-H., and Lin, Y.-Y., "Deep semantic matching with foreground detection and cycle-consistency," in *Asian Conference on Computer Vision*, pp. 347–362, Springer, 2018.

[22] Chen, Z., Jacobson, A., Sünderhauf, N., Upcroft, B., Liu, L., Shen, C., Reid, I., and Milford, M., "Deep learning features at scale for visual place recognition," in *IEEE International Conference on Robotics and Automation*, pp. 3223–3230, 2017.

[23] Churchill, W. and Newman, P., "Experience-based navigation for long-term localisation," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[24] Clayton, N. S. and Krebs, J. R., "Memory for spatial and object-specific cues in food-storing and non-storing birds," *Journal of Comparative Physiology A*, vol. 174, no. 3, pp. 371–379, 1994.

[25] Corke, P., Paul, R., Churchill, W., and Newman, P., "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2085–2092, 2013.

[26] Csurka, G., ed., *Domain Adaptation in Computer Vision Applications*. Springer, 2017.

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

[28] De Kort, S. R. and Clayton, N. S., "An evolutionary perspective on caching by corvids," *Royal Society of London B: Biological Sciences*, vol. 273, no. 1585, pp. 417–423, 2006.

[29] Dellaert, F., "Factor Graphs and GTSAM: A Hands-on Introduction," Tech. Rep. GT-RIM-CP&R-2012-002, GT RIM, Sept 2012.

[30] DeTone, D., Malisiewicz, T., and Rabinovich, A., "Toward geometric deep slam," *arXiv preprint arXiv:1707.07410*, 2017.

[31] DeTone, D., Malisiewicz, T., and Rabinovich, A., "Superpoint: Self-supervised interest point detection and description," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 224–236, 2018.

[32] Diego, F., Ponsa, D., Serrat, J., and López, A. M., "Video alignment for change detection," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1858–1869, 2011.

[33] Dong, J., Boots, B., Dellaert, F., Chandra, R., and Sinha, S., "Learning to align images using weak geometric supervision," in *IEEE International Conference on 3D Vision*, pp. 700–709, 2018.

[34] Dong, J., Burnham, J. G., Boots, B., Rains, G., and Dellaert, F., "4D crop monitoring: Spatio-temporal reconstruction for agriculture," in *IEEE International Conference on Robotics and Automation*, pp. 3878–3885, 2017.

[35] Dong, J., Nelson, E., Indelman, V., Michael, N., and Dellaert, F., "Distributed real-time cooperative localization and mapping using an uncertainty-aware expectation maximization approach," in *IEEE International Conference on Robotics and Automation*, pp. 5807–5814, 2015.

[36] Durmush, A., Suominen, O., Yli-Hietanen, J., Peltonen, S., Collin, J., and Gotchev, A., "FinnForest: A forest landscape for visual slam." https://etsin.fairdata.fi/dataset/bacdcc39-62df-455c-947e-ff343f467dab, October 2019.

[37] Dvijotham, K., Gowal, S., Stanforth, R., Arandjelovic, R., O'Donoghue, B., Uesato, J., and Kohli, P., "Training verified learners with learned verifiers," *arXiv preprint arXiv:1805.10265*, 2018.

[38] Dymczyk, M., Lynen, S., Cieslewski, T., Bosse, M., Siegwart, R., and Furgale, P., "The gist of maps-summarizing experience for lifelong localization," in *IEEE International Conference on Robotics and Automation*, pp. 2767–2773, 2015.

[39] Engel, J., Koltun, V., and Cremers, D., "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[40] Engel, J., Schöps, T., and Cremers, D., "LSD-SLAM: Large-scale direct monocular SLAM," in *IEEE European Conference on Computer Vision*, pp. 834–849, 2014.

[41] Evangelidis, G. D. and Bauckhage, C., "Efficient subframe video alignment using short descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2371–2386, 2013.

[42] Facil, J. M., Olid, D., Montesano, L., and Civera, J., "Condition-invariant multi-view place recognition," *arXiv preprint arXiv:1902.09516*, 2019.

[43] Ferguson, D., Morris, A., Haehnel, D., Baker, C., Omohundro, Z., Reverte, C., Thayer, S., Whittaker, C., Whittaker, W., Burgard, W., and others, "An autonomous robotic system for mapping abandoned mines," in *Neural Information Processing Systems*, pp. 587–594, 2004.

[44] Furgale, P., Carle, P., Enright, J., and Barfoot, T. D., "The devon island rover navigation dataset," *International Journal of Robotics Research*, vol. 31, no. 6, pp. 707–713, 2012.

[45] Garforth, J. and Webb, B., "Visual appearance analysis of forest scenes for monocular slam," in *IEEE International Conference on Robotics and Automation*, pp. 1794–1800, 2019.

[46] Giusti, A., Guzzi, J., Ciresan, D., He, F.-L., Rodriguez, J. P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., and others, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, 2015.

[47] Glover, A., Maddern, W., Milford, M., and Wyeth, G., "FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day," in *IEEE International Conference on Robotics and Automation*, 2010.

[48] Glover, A. J., Maddern, W. P., Milford, M. J., and Wyeth, G. F., "FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day," in *IEEE International Conference on Robotics and Automation*, pp. 3507–3512, 2010.

[49] Gomez-Ojeda, R., Lopez-Antequera, M., Petkov, N., and Gonzalez-Jimenez, J., "Training a convolutional neural network for appearance-invariant place recognition," *arXiv*, 2015. arXiv:1505.07428.

[50] Gonzales, D. and Brown, S., "Symbiosis: A surprising tale of species cooperation." `http://ed.ted.com/lessons/symbiosis-a-surprising-tale-of-species-cooperation`, 2012. Accessed: 2016-04-27.

[51] Gout, A., Lifchitz, Y., Cottencin, T., Groshens, Q., Fix, J., and Pradalier, C., "Evaluation of off-the-shelf cnns for the representation of natural scenes with large seasonal variations," 2017.

[52] Graham, M. C., How, J. P., and Gustafson, D. E., "Robust incremental slam with consistency-checking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 117–124, 2015.

[53] Griffith, S., Chahine, G., and Pradalier, C., "Symphony lake dataset," *International Journal of Robotics Research*, vol. 36, pp. 1151–1158, 2017.

[54] Griffith, S., Dellaert, F., and Pradalier, C., "Robot-Enabled Lakeshore Monitoring Using Visual SLAM and SIFT Flow," in *RSS Workshop on Multi-View Geometry in Robotics*, 2015.

[55] Griffith, S., Dellaert, F., and Pradalier, C., "Transforming Multiple Visual Surveys of a Natural Environment Into Time–Lapses," *International Journal of Robotics Research*, 2019.

[56] Griffith, S., Drews, P., and Pradalier, C., "Towards autonomous lakeshore monitoring," in *International Symposium on Experimental Robotics*, 2014.

[57] Griffith, S. and Pradalier, C., "A spatially and temporally scalable approach for long-term lakeshore monitoring," in *Conference On Field And Service Robotics*, 2015.

[58] Griffith, S. and Pradalier, C., "Reprojection flow for image registration across seasons," in *British Machine Vision Conference*, 2016.

[59] Griffith, S. and Pradalier, C., "Survey registration for long-term natural environment monitoring," *Journal of Field Robotics*, vol. 34, no. 1, pp. 188–208, 2017.

[60] Griffith, S., Sukhoy, V., and Stoytchev, A., "Using sequences of movement dependency graphs to form object categories," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 715–720, 2011.

[61] Gu, J., Ramamoorthi, R., Belhumeur, P., and Nayar, S., "Removing image artifacts due to dirty camera lenses and thin occluders," *ACM Transactions on Graphics*, vol. 28, no. 5, p. 144, 2009.

[62] Ham, B., Cho, M., Schmid, C., and Ponce, J., "Proposal flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3475–3484, 2016.

[63] Hartley, R. I. and Sturm, P., "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[64] He, K., Lu, Y., and Sclaroff, S., "Local descriptors optimized for average precision," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 596–605, 2018.

[65] He, X., Zemel, R., and Mnih, V., "Topological map learning from outdoor image sequences," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1091–1104, 2006.

[66] Heidarsson, H. and Sukhatme, G., "Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3160–3165, 2011.

[67] Hinton, G. E. and Salakhutdinov, R. R., "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[68] Hitz, G., Gotovos, A., Pomerleau, F., Garneau, M.-E., Pradalier, C., Krause, A., and Siegwart, R. Y., "Fully autonomous focused exploration for robotic environmental monitoring," in *IEEE International Conference on Robotics and Automation*, pp. 2658–2664, 2014.

[69] Hitz, G., Pomerleau, F., Colas, F., and Siegwart, R., "State estimation for shore monitoring using an autonomous surface vessel," in *International Symposium on Experimental Robotics*, 2014.

[70] Hitz, G., Pomerleau, F., Garneau, M.-E., Pradalier, C., Posch, T., Pernthaler, J., and Siegwart, R. Y., "Autonomous inland water monitoring: Design and application of a surface vessel," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 62–72, 2012.

[71] Hutchins, H. E. and Lanner, R. M., "The central role of clark's nutcracker in the dispersal and establishment of whitebark pine," *Oecologia*, vol. 55, no. 2, pp. 192–201, 1982.

[72] Indelman, V., Nelson, E., Michael, N., and Dellaert, F., "Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization," in *IEEE International Conference on Robotics and Automation*, pp. 593–600, 2014.

[73] Jacobs, N., Burgin, W., Fridrich, N., Abrams, A., Miskell, K., Braswell, B. H., Richardson, A. D., and Pless, R., "The global network of outdoor webcams: properties and applications," in *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 111–120, 2009.

[74] Jain, S., Nuske, S. T., Chambers, A. D., Yoder, L., Cover, H., Chamberlain, L. J., Scherer, S., and Singh, S., "Autonomous river exploration," in *Conference on Field and Service Robotics*, December 2013.

[75] Jeon, S., Min, D., Kim, S., and Sohn, K., "Joint learning of semantic alignment and object landmark detection," *arXiv preprint arXiv:1910.00754*, 2019.

[76] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T., "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, pp. 675–678, 2014.

[77] Johannsson, H., Kaess, M., Fallon, M., and Leonard, J. J., "Temporally scalable visual slam using a reduced pose graph," in *IEEE International Conference on Robotics and Automation*, pp. 54–61, 2013.

[78] Johns, E. and Yang, G.-Z., "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *IEEE International Conference on Robotics and Automation*, pp. 3212–3218, 2013.

[79] Johnson-Roberson, M., Pizarro, O., Williams, S. B., and Mahon, I., "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys," *Journal of Field Robotics*, vol. 27, no. 1, pp. 21–51, 2010.

[80] Jones, E. S. and Soatto, S., "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011.

[81] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F., "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[82] Kendall, A., Gal, Y., and Cipolla, R., "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018.

[83] Kim, J., Liu, C., Sha, F., and Grauman, K., "Deformable spatial pyramid matching for fast dense correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2307–2314, 2013.

[84] Kim, S., Min, D., Ham, B., Jeon, S., Lin, S., and Sohn, K., "FCSS: Fully convolutional self-similarity for dense semantic correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6560–6569, 2017.

[85] Kim, S., Min, D., Lin, S., and Sohn, K., "DCTM: Discrete-continuous transformation matching for semantic flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4529–4538, 2017.

[86] Kopf, J., Cohen, M. F., and Szeliski, R., "First-person hyper-lapse videos," *ACM Transactions on Graphics*, vol. 33, no. 4, p. 78, 2014.

[87] Košecka, J., "Detecting changes in images of street scenes," in *Computer Vision–ACCV 2012*, vol. 7727 of *LNCS*, pp. 590–601, Springer, 2013.

[88] Krajnik, T., Cristóforis, P., Nitsche, M., Kusumam, K., and Duckett, T., "Image features and seasons revisited," in *IEEE European Conference on Mobile Robots*, pp. 1–7, 2015.

[89] Krebs, J. R., Clayton, N. S., Healy, S. D., Cristol, D. A., Patel, S. N., and Jolliffe, A. R., "The ecology of the avian brain: Food-storing memory and the hippocampus," *International Journal of Avian Science*, vol. 138, no. 1, pp. 34–46, 1996.

[90] Kularatne, D. and Hsieh, A., "Tracking attracting lagrangian coherent structures in flows," in *Robotics: Science and Systems*, 2015.

[91] Kulkarni, N., Gupta, A., and Tulsiani, S., "Canonical surface mapping via geometric cycle consistency," *arXiv preprint arXiv:1907.10043*, 2019.

[92] Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A., "Graphcut textures: Image and video synthesis using graph cuts," in *ACM Transactions on Graphics*, vol. 22, pp. 277–286, 2003.

[93] Laffont, P.-Y., Ren, Z., Tao, X., Qian, C., and Hays, J., "Transient attributes for high-level understanding and editing of outdoor scenes," *ACM Transactions on Graphics*, vol. 33, no. 4, p. 149, 2014.

[94] Lai, H.-Y., Tsai, Y.-H., and Chiu, W.-C., "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1890–1899, 2019.

[95] Laskar, Z., Melekhov, I., Tavakoli, H. R., Ylioinas, J., and Kannala, J., "Geometric image correspondence verification by dense pixel matching," *arXiv preprint arXiv:1904.06882*, 2019.

[96] Latif, Y., Cadena, C., and Neira, J., "Robust loop closing over time for pose graph slam," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, 2013.

[97] Le, H. and Milford, M., "Large scale visual place recognition with sub-linear storage growth," *arXiv preprint arXiv:1810.09660*, 2018.

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

[99] Linegar, C., Churchill, W., and Newman, P., "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation," in *IEEE International Conference on Robotics and Automation*, pp. 90–97, 2015.

[100] Liu, C., Yuen, J., and Torralba, A., "SIFT Flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[101] Long, J. L., Zhang, N., and Darrell, T., "Do convnets learn correspondence?," in *Neural Information Processing Systems*, pp. 1601–1609, 2014.

[102] Lopez-Antequera, M., Gomez-Ojeda, R., Petkov, N., and Gonzalez-Jimenez, J., "Appearance-invariant place recognition by discriminatively training a convolutional neural network," *Pattern Recognition Letters*, vol. 92, pp. 89–95, 2017.

[103] Lowe, D., "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[104] Lowry, S. and Milford, M. J., "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[105] Lowry, S., Sunderhauf, N., Newman, P., Leonard, J. J., Cox, D., Corke, P., and Milford, M. J., "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 1–19, 2016.

[106] Lucas, B. D. and Kanade, T., "An iterative image registration technique with an application to stereo vision.," in *International Joint Conference on Artificial Intelligence*, vol. 81, pp. 674–679, 1981.

[107] Ma, W.-C., Wang, S., Hu, R., Xiong, Y., and Urtasun, R., "Deep rigid instance scene flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3614–3622, 2019.

[108] Maddern, W., Pascoe, G., Linegar, C., and Newman, P., "1 year, 1000 km: The Oxford RobotCar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[109] Maddern, W., Stewart, A., McManus, C., Upcroft, B., Churchill, W., and Newman, P., "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *IEEE ICRA Workshop on Visual Place Recognition in Changing Environments Workshop*, 2014.

[110] Martin-Brualla, R., Gallup, D., and Seitz, S. M., "3d time-lapse reconstruction from internet photos," in *IEEE International Conference on Computer Vision*, pp. 1332–1340, 2015.

[111] Martin-Brualla, R., Gallup, D., and Seitz, S. M., "Time-lapse mining from internet photos," *ACM Transactions on Graphics*, vol. 34, no. 4, p. 62, 2015.

[112] MAWI United GmbH. `https://www.unrealengine.com/marketplace/en-US/profile/MAWI+United+GmbH`, 2019. High-end CGI for UE4 and VFX.

[113] McDonald, J., Kaess, M., Cadena, C., Neira, J., and Leonard, J. J., "Real-time 6-dof multi-session visual slam over large-scale environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1144–1158, 2013.

[114] McManus, C., Upcroft, B., and Newman, P., "Scene signatures: Localized and point-less features for localization," in *Robotics: Science and Systems*, 2014.

[115] Melekhov, I., Tiulpin, A., Sattler, T., Pollefeys, M., Rahtu, E., and Kannala, J., "Dgc-net: Dense geometric correspondence network," in *IEEE Winter Conference on Applications of Computer Vision*, pp. 1034–1042, 2019.

[116] Milford, M., "Vision-based place recognition: How low can you go?," *International Journal of Robotics Research*, vol. 32, no. 7, pp. 766–789, 2013.

[117] Milford, M., Firn, J., Beattie, J., Jacobson, A., Pepperell, E., Mason, E., Kimlin, M., and Dunbabin, M., "Automated sensory data alignment for environmental and epidermal change monitoring," in *Australasian Conference on Robotics and Automation 2014*, pp. 1–10, Australian Robotic and Automation Association, 2014.

[118] Milford, M. and Wyeth, G., "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*, pp. 1643–1649, 2012.

[119] Milford, M. J. and Wyeth, G. F., "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation*, pp. 1643–1649, 2012.

[120] Min, J., Lee, J., Ponce, J., and Cho, M., "Hyperpixel flow: Semantic correspondence with multi-layer neural features," in *IEEE International Conference on Computer Vision*, 2019.

[121] Mühlfellner, P., Bürki, M., Bosse, M., Derendarz, W., Philippsen, R., and Furgale, P., "Summary maps for lifelong visual localization," *Journal of Field Robotics*, 2015.

[122] Naseer, T., Burgard, W., and Stachniss, C., "Robust visual localization across seasons," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 289–302, 2018.

[123] Naseer, T., Oliveira, G. L., Brox, T., and Burgard, W., "Semantics-aware visual localization under challenging perceptual conditions," in *IEEE International Conference on Robotics and Automation*, 2017.

[124] Naseer, T., Ruhnke, M., Stachniss, C., Spinello, L., and Burgard, W., "Robust visual slam across seasons," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[125] Neubert, P., Sunderhauf, N., and Protzel, P., "Appearance change prediction for long-term navigation across seasons," in *IEEE European Conference on Mobile Robots*, pp. 198–203, 2013.

[126] Ni, K., Steedly, D., and Dellaert, F., "Tectonic SAM: Exact, out-of-core, submap-based SLAM," in *IEEE International Conference on Robotics and Automation*, pp. 1678–1685, 2007.

[127] Niinioja, R., Holopainen, A.-L., Lepistö, L., Rämö, A., and Turkka, J., "Public participation in monitoring programmes as a tool for lakeshore monitoring: The example of Lake Pyhäjärvi, Karelia, Eastern Finland," *Limnologica-Ecology and Management of Inland Waters*, vol. 34, no. 1, pp. 154–159, 2004.

[128] Olid, D., Fácil, J. M., and Civera, J., "Single-view place recognition under seasonal changes," in *IEEE/RSJ IROS Workshop on Planning, Perception, and Navigation for Intelligent Vehicles*, 2018.

[129] Oliva, A. and Torralba, A., "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[130] Olson, E. and Agarwal, P., "Inference on networks of mixtures for robust robot mapping," *International Journal of Robotics Research*, vol. 32, no. 7, pp. 826–840, 2013.

[131] Pashler, H., "Familiarity and visual change detection," *Perception & psychophysics*, vol. 44, no. 4, pp. 369–378, 1988.

[132] Pavlovic, N. B. and Bowles, M. L., "Rare plant monitoring at indiana dunes national lakeshore," *Science and ecosystem management in the national parks. University of Arizona Press, Tucson*, pp. 253–280, 1996.

[133] Pedersen, L., Smith, T., Lee, S. Y., and Cabrol, N., "Planetary lakelander–a robotic sentinel to monitor remote lakes," *Journal of Field Robotics*, vol. 32, no. 6, pp. 860–879, 2015.

[134] Pfingsthorn, M. and Birk, A., "Generalized graph slam: Solving local and global ambiguities through multimodal and hyperedge constraints," *International Journal of Robotics Research*, vol. 35, no. 6, pp. 601–630, 2016.

[135] Pradalier, C., Aravecchia, S., and Pomerleau, F., "Multi-session lakeshore monitoring in visually challenging conditions," in *Conference on Field and Service Robotics*, 2019.

[136] Radenović, F., Tolias, G., and Chum, O., "Fine-tuning cnn image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[137] Revaud, J., Almazan, J., de Rezende, R. S., and de Souza, C. R., "Learning with average precision: Training image retrieval with a listwise loss," *arXiv preprint arXiv:1906.07589*, 2019.

[138] Revaud, J., Weinzaepfel, P., De Souza, C., Pion, N., Csurka, G., Cabon, Y., and Humenberger, M., "R2D2: Repeatable and Reliable Detector and Descriptor," *arXiv preprint arXiv:1906.06195*, 2019.

[139] Revaud, J., Weinzaepfel, P., Harchaoui, Z., and Schmid, C., "Deepmatching: Hierarchical deformable dense matching," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323, 2016.

[140] Rocco, I., Arandjelovic, R., and Sivic, J., "Convolutional neural network architecture for geometric matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6148–6157, 2017.

[141] Rocco, I., Arandjelović, R., and Sivic, J., "End-to-end weakly-supervised semantic alignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6917–6925, 2018.

[142] Rocco, I., Cimpoi, M., Arandjelović, R., Torii, A., Pajdla, T., and Sivic, J., "Neighbourhood consensus networks," in *Neural Information Processing Systems*, pp. 1651–1662, 2018.

[143] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "Orb: an efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.

[144] Rusu, R. B. and Cousins, S., "3D is here: Point cloud library (pcl)," in *IEEE International Conference on Robotics and Automation*, 2011.

[145] Sakamoto, T. and Yasuda, N., "Monitoring and evaluating dams and reservoirs," *Water Storage, Transport, and Distribution*, p. 176, 2009.

[146] Sand, P. and Teller, S., "Video matching," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 592–599, 2004.

[147] Sarlin, P.-E., Cadena, C., Siegwart, R., and Dymczyk, M., "From coarse to fine: Robust hierarchical localization at large scale," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019.

[148] Sattler, T., Leibe, B., and Kobbelt, L., "Fast image-based localization using direct 2D-to-3D matching," in *IEEE International Conference on Computer Vision*, pp. 667–674, 2011.

[149] Scott, J. D., *Clark's nutcracker occurrence, whitebark pine stand health, and cone production in the waterton-glacier international peace park*. PhD thesis, University of Colorado, 2013.

[150] Sibley, G., Mei, C., Reid, I., and Newman, P., "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *International Journal of Robotics Research*, vol. 29, no. 8, pp. 958–980, 2010.

[151] Sivic, J. and Zisserman, A., "Video Google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, pp. 1470–1477, 2003.

[152] Skeele, R. C. and Hollinger, G. A., "Aerial vehicle path planning for monitoring wildfire frontiers," in *Conference on Field and Service Robotics*, pp. 455–467, Springer, 2016.

[153] Skrede, S., "Nordlandsbanen: minute by minute, season by season." `https://nrkbeta.no/2013/01/15/nordlandsbanen-minute-by-minute-season-by-season/`, January 2013.

[154] Stoytchev, A., "Some basic principles of developmental robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 2, pp. 122–130, 2009.

[155] Stumm, E., Mei, C., and Lacroix, S., "Probabilistic place recognition with covisibility maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4158–4163, 2013.

[156] Subramanian, A., Gong, X., Riggins, J., Stilwell, D., and Wyatt, C., "Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications," in *IEEE OCEANS*, pp. 1–6, 2006.

[157] Sünderhauf, N. and Protzel, P., "Switchable constraints for robust pose graph slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1879–1884, 2012.

[158] Sunderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M., "On the performance of convnet features for place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4297–4304, 2015.

[159] Sunderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., and Milford, M., "Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free," *Robotics: Science and Systems*, 2015.

[160] Sutton, R., "Verification: The Key to AI." http://webdocs.cs.ualberta.ca/ sutton/IncIdeas/KeytoAI.html, Nov 2001.

[161] Tang, J., Ericson, L., Folkesson, J., and Jensfelt, P., "Gcnv2: Efficient correspondence prediction for real-time slam," *IEEE Robotics and Automation Letters*, 2019.

[162] Thrun, S., Thayer, S., Whittaker, W., Baker, C., Burgard, W., Ferguson, D., Hähnel, D., Montemerlo, M., Morris, A., Omohundro, Z., Reverte, C., and Whittaker, W., "Autonomous exploration and mapping of abandoned mines," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 79–91, 2004.

[163] Tokekar, P., Branson, E., Vander Hook, J., and Isler, V., "Tracking aquatic invaders: Autonomous robots for monitoring invasive fish," *IEEE Robotics and Automation Magazine*, vol. 20, no. 3, pp. 33–41, 2013.

[164] Tomback, D. F., "Foraging strategies of clark's nutcracker," *Living Bird*, vol. 16, no. 1977, pp. 123–160, 1978.

[165] Valada, A., Oliveira, G., Brox, T., and Burgard, W., "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *International Symposium on Experimental Robotics*, 2016.

[166] von Stumberg, L., Wenzel, P., Khan, Q., and Cremers, D., "Gn-net: The gauss-newton loss for multi-weather relocalization," *preprint*, 2019.

[167] Vysotska, O. and Stachniss, C., "Lazy data association for image sequences matching under substantial appearance changes," in *IEEE International Conference on Robotics and Automation*, pp. 213–220, 2016.

[168] Wang, O., Schroers, C., Zimmer, H., Gross, M., and Sorkine-Hornung, A., "VideoSnapping: Interactive Synchronization of Multiple Videos," *ACM Transactions on Graphics*, vol. 33, pp. 77:1–77:10, July 2014.

[169] Wu, X. and Pradalier, C., "Multi-scale direct sparse visual odometry for large-scale natural environment," in *IEEE International Conference on 3D Vision*, pp. 89–97, 2018.

[170] Yang, H., Lin, W.-Y., and Lu, J., "Daisy Filter Flow: A generalized discrete approach to dense correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3406–3413, 2014.

[171] Yang, Y., Wong, A., and Soatto, S., "Dense depth posterior (ddp) from single image and sparse range," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3353–3362, 2019.

[172] Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., and Oliva, A., "Places: An image database for deep scene understanding," *Journal of Vision*, vol. 17, no. 10, 2016.

[173] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A., "Learning deep features for scene recognition using places database," in *Neural Information Processing Systems* (Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., eds.), pp. 487–495, 2014.

[174] Zhou, T., Krahenbuhl, P., Aubry, M., Huang, Q., and Efros, A. A., "Learning dense correspondence via 3d-guided cycle consistency," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 117–126, 2016.

[175] Zhou, T., Lee, Y. J., Yu, S. X., and Efros, A. A., "FlowWeb: Joint image set alignment by weaving consistent, pixel-wise correspondences," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1191–1200, 2015.