

A Spatially and Temporally Scalable Approach for Long-Term Lakeshore Monitoring

Shane Griffith and Cédric Pradalier

Abstract This paper provides an image processing framework to assist in the inspection and, more generally, the data association of a natural environment, which we demonstrate in a long-term lakeshore monitoring task with an autonomous surface vessel. Our domain consists of 55 surveys of a 1 km lakeshore collected over a year and a half. Our previous work introduced a framework in which images of the same scene from different surveys are aligned using visual SLAM and SIFT Flow. This paper: 1) minimizes the number of expensive image alignments between two surveys using a covering set of poses, rather than all the poses in a sequence; 2) improves alignment quality using a local search around each pose and an alignment bias derived from the 3D information from visual SLAM; and 3) provides exhaustive results of image alignment quality. Our improved framework finds significantly more precise alignments despite performing image registration over an order of magnitude fewer times. We show changes a human spotted between surveys that would have otherwise gone unnoticed. We also show cases where our approach was robust to ‘extreme’ variation in appearance.

1 Introduction

This paper presents an application of autonomous surface vessels (ASV) for long-term observation of lakeshore environments. A growing number of robots are being targeted for inspection tasks in natural environments, including applications in agriculture, surveillance, and environment monitoring. Yet, the variation of appearance of outdoor environments significantly limits robots in tasks requiring data associa-

Shane Griffith
Georgia Institute of Technology. e-mail: sgriffith7@gatech.edu

Cédric Pradalier
GeorgiaTech Lorraine - UMI 2958 GT-CNRS. e-mail: cedric.pradalier@gatech.edu

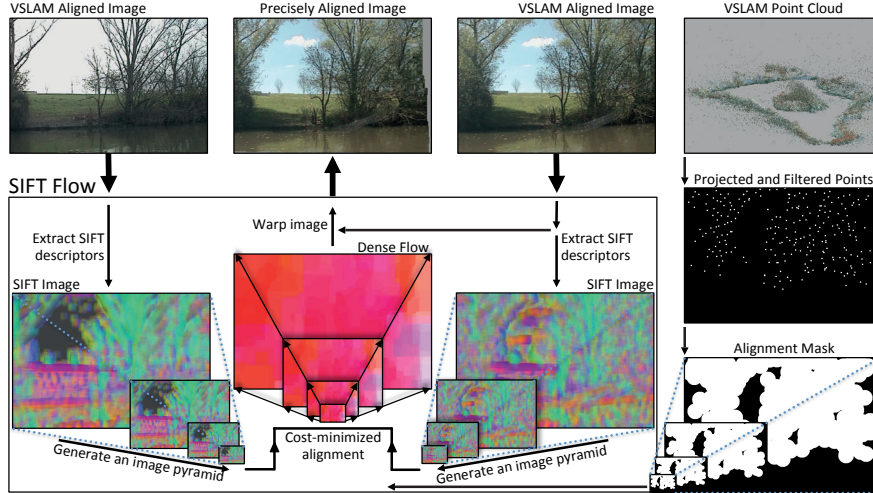


Fig. 1 The registration of two images. For each VSLAM aligned image, SIFT descriptors are computed at each pixel to form a SIFT image, which is down-sampled into an image pyramid. To avoid aligning noise due to the sky and the water, the alignment cost is biased using an image mask of the lakeshore (derived from the 3D information in the feature tracks of visual SLAM). The resulting dense flow aligns the two input images, which enables quick change detection for manual inspection tasks.

tion, with many papers only addressing particular aspects of the variation (e.g., illumination/shadows [4], night [19], and noise [8]). Scene structure can help provide robustness to natural variation of appearance [16, 18]. Few papers have, however, demonstrated robust data association across surveys of natural environments.

This paper presents a framework for achieving high resolution, pixel-level alignment between fortnightly surveys of a lakeshore. Our framework uses visual SLAM (see e.g., [1, 21, 22]) to identify images of roughly the same scene from different surveys and then it applies SIFT Flow [16] to precisely align them (see Fig. 1). Building on our previous work [6, 7], in this paper we minimize the number of expensive image alignments using a covering set of poses. To improve image alignment accuracy at a particular pose, a search for the best alignment is performed in its tight neighborhood of images. Image alignment accuracy is further improved using the 3D landmark positions from visual SLAM to bias the image registration process. Once images are precisely aligned, a human inspecting them can easily spot if something changed. At this stage, given the difficulty of automatically processing natural scenes, we are assuming the inspection task is left to a human, but we endeavor to make his/her task as efficient as possible.

To date, we have surveyed a lake a total of 55 times over 1.5 years, which represents a spatially large and a temporally long scale study using ASVs. We show our framework enabled a human to detect changes that would have otherwise gone unnoticed. We also show our approach is robust to variation in appearance of the sky,

the water, changes in objects on a lakeshore, and the seasonal changes of plants. Finally, we point out failure cases, which indicate directions for future work.

2 Related Work

The field of Simultaneous Localization and Mapping (SLAM) provides a foundation for localizing a robot and mapping monitored spaces. Gaining the advantages of SLAM in natural environments requires, however, a system made to handle three particular challenges: 1) the large spatial scale; 2) the non-rigid environments (e.g., moving trees, changing water levels); and 3) the very high level of visual similarity (e.g., branches and leaves of different trees may appear to be from the same one). The variation of appearance over a long-term monitoring task further increases the difficulty of data association in surveys of an outdoor environment.

Many different techniques have been proposed to solve data association for outdoor environments. Some approaches rely on point-based features such as SIFT (e.g., [14, 1, 9]) for performing data association. Point-based feature matching is, however, often not robust to common sources of variation (see e.g., [6]). In light of this, some work has focused on directly using, or modifying, whole or parts of images. Neubert et al. [20] deals with seasonal changes by introducing a prediction step in which whole images are modified to look more like the current season. McManus et al. [18] utilize patches of images, called ‘scene signatures’, which are matched using classifiers and capture information about the structure of each scene. In case a particular location is stubborn to feature- and whole-image-based data association, ‘multiple experiences’ of the location can be accumulated until new observations are associated well [2]. This paper performs data association using SIFT Flow [16], an algorithm designed to find dense correspondences using whole images worth of point-based features. It combines the accuracy of point-based feature matching with the robustness of whole-image matching.

Traversing a lake while mapping the location of a lakeshore is an essential task of lakeshore monitoring, which some papers have already started to address. Heidarsson and Sukhatme [10] and Subramanian et al. [23] map a lakeshore and the locations of obstacles from the visual perspectives of their ASVs. Jain et al. [12] proposed to use a drone for autonomously mapping riverine environments, which can avoid debris in the water, yet fly below dense tree cover. In case a robot repeatedly visits the same lakeshore, Hitz et al. [11] show that 3D laser scans of a shoreline can be used to delineate 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 spring. This paper combines iSAM2 [13] for scalable SLAM with SIFT Flow for robust data association in a framework for long-term lakeshore monitoring.

3 Experimental Setup

We used Clearpath’s Kingfisher ASV for our experiments. It is 1.35 meters long and 0.98 meters wide, with two pontoons, a water-tight compartment to house electronics, and an area on top for sensors and the battery. It is propelled by a water jet in each of its pontoons and turns during power differentials. It can reach a top speed of about 2 m/s, but we mostly operated it at lower speeds to maximize battery life, which is about an hour with our current payload.

Our Kingfisher is equipped with a suite of exteroceptive and interoceptive sensors. A 704x480 color pan-tilt camera captures images at 10 frames per second. Beneath it sits a single scan line laser rangefinder with a field of view of about 270 degrees. It is pointed just above the surface of the water and provides a distance estimate for everything less than 20m away. The watertight compartment houses a GPS, a compass, and an IMU.

The ASV was deployed on Symphony Lake in Metz, France, which is about 400 meters long and 200 meters wide with an 80 meter-wide island in the middle. The nature of the lakeshore varied, with shrubs, trees, boulders, grass, sand, buildings, birds, and people in the immediate surroundings. People mostly kept to the walking trail and a bike path a few meters from the shore, and fishermen occasionally sat along the shore.

We used a simple set of behaviors to autonomously steer the robot around the perimeter of the lake and the island. As the boat moves at a constant velocity of about 0.4m/s, a local planner chooses among a set of state lattice motion primitives to keep the boat 10m away from the lakeshore on its starboard side. With this configuration, the robot is capable of performing an entire survey autonomously; however, we occasionally took control using a remote control in order to avoid fishing lines, debris, to swap batteries, etc.

We have regularly deployed the robot up to once per week since August 18, 2013. This paper analyzes data from 10 different surveys, which span seven months of variation. All 10 were chosen because they each consisted of one run around the entire lakeshore, including the island. Each survey was performed in the daytime on a weekday in sunny or cloudy weather, at various times of the day.

4 Methodology

Our framework aligns images between two surveys using a coarse-to-fine process with four main components. First visual SLAM is used to localize the trajectory of the ASV and map visual features of the shore. Second a minimum view set is identified, which covers all the sections of lakeshore with similar viewpoints in both surveys. Third, given two poses facing the same scene from two different surveys, a process of image registration using SIFT Flow is performed for the best pixel-wise alignment. In the last step images are presented to an end user in a flickering display.

A single survey represents a collection of image sequences, measurements of the camera pose, and other useful information about the robot’s movement. During a survey, k , the robot acquires the tuple $\mathcal{A}^k = \{\mathcal{T}_i^k, \mathcal{I}_i^k, \hat{\mathbf{C}}_i^k, \hat{\boldsymbol{\omega}}_i^k\}_{i=1}^{|\mathcal{A}^k|}$ every 10^{th} of a second, where \mathcal{T} is the current time, \mathcal{I} is the image from the pan-tilt camera, $\hat{\mathbf{C}} \in \text{SE}(3)$ is the estimated camera pose, and $\hat{\boldsymbol{\omega}}$ is the estimated angular velocity of the boat. The estimated camera pose is derived from the boat’s GPS position, the boat’s compass heading, and the pan and tilt positions of the camera. The IMU provides $\hat{\boldsymbol{\omega}}$. Each survey is down-sampled by a factor of five to reduce data redundancy and speed up computation time.

Finding nearby images in two long surveys is possible using raw measurements of the camera pose, but because these measurements are prone to noise that could lead to trying to align images of two different scenes, we begin by using visual-inertial SLAM to improve our estimates of the camera poses.

4.1 Visual SLAM

We used generic feature tracking for visual SLAM, which is based on detecting 300 Harris corner features and then tracking them using the pyramidal Lucas–Kanade Optical Flow algorithm (from OpenCV) as the boat moves. We then apply a graph-based SLAM approach for optimizing the camera poses and the visual feature locations. A factor graph is used to represent the set of measurements of the camera poses and the landmark positions, and the different constraints between them. The GTSAM bundle adjustment framework is applied to the factor graph to reduce the error in the initial estimates of the positions [5]. See [7] for more details.

4.2 Selecting a Minimum View Set

To reduce the computational overhead of image alignment (Section 4.3) and to enable a manual comparison between two surveys (Section 4.4), we select a minimum view set from among the roughly 50,000 images of each survey. A large set of images in each survey is desirable for the optical flow step of visual SLAM and to reduce motion blur. Yet, it means the images have a lot of redundancy, which is cumbersome for a survey comparison. Ideally, a person comparing two surveys would only see a subset of these images, where each corresponds to a unique section of the shore. This section describes how we find a minimal subset of images that covers as much lakeshore as is seen in both surveys.

Another name for this is the “Set Cover Problem” (SCP) [3], which can be expressed as follows. Let \mathcal{S} be the set of all the observable positions in a survey of a lakeshore. Each camera pose, i , of the survey observes a subset \mathcal{I}_i of these shore points, where $\mathcal{S} = \bigcup_{i \in I} \mathcal{I}_i$. The goal is to find a set of poses J for which $\mathcal{S} = \bigcup_{j \in J} \mathcal{I}_j$ and $|J|$ is as small as possible. This Set Cover Problem is NP-Hard. It can be approximated using linear programming or a simple greedy approach, which gives sufficient performance for our application.

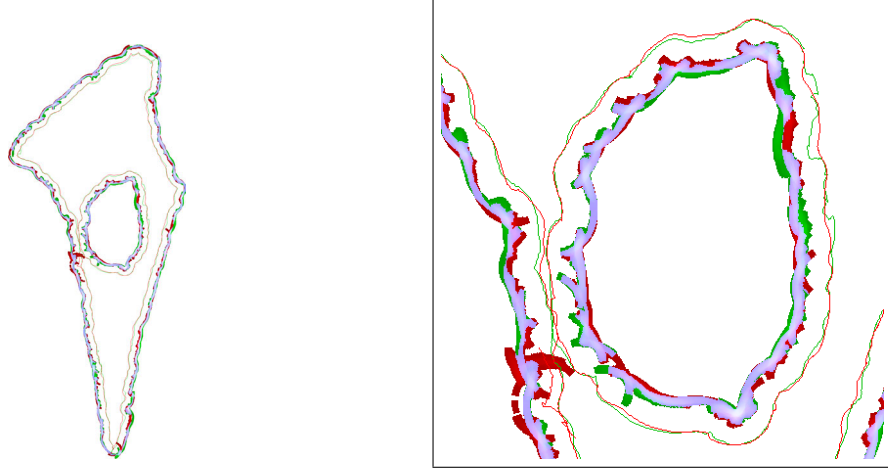


Fig. 2 The recorded trajectory of the boat and the shore points it sees for two surveys. The shore points seen from the red trajectory are displayed in red, those seen from the green trajectory are green, and those seen from both are mauve. A closeup of the island is shown on the right.

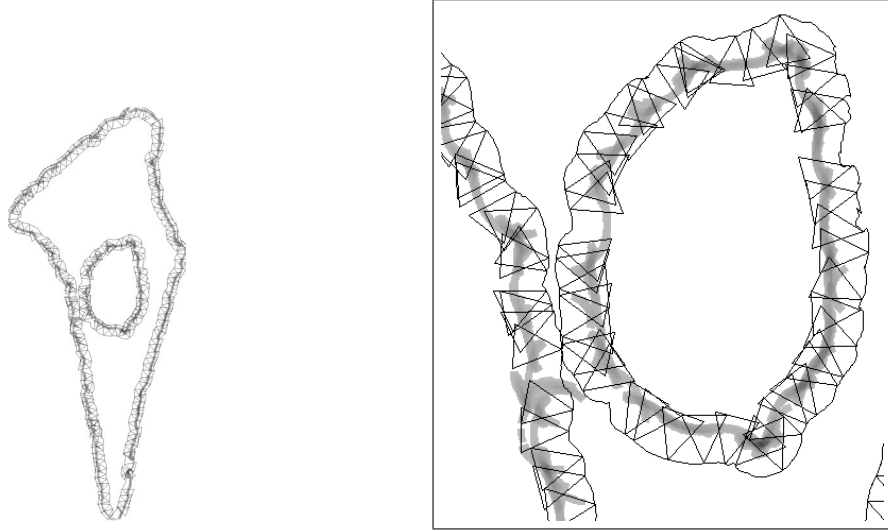


Fig. 3 The cover set for the red survey from Fig. 2, which accounts for the estimated co-visibility with the green survey. Black triangles indicate the visibility frustum of the selected images. A closeup of the island is shown on the right.

The set of shore points that compose \mathcal{S} is identified using the optimized poses from visual SLAM. Because the robot is controlled to move at a constant distance, d , from the shore, every point $d \pm \varepsilon$ away in the camera frustum is considered part of it (where $d = 10\text{m}$ and $\varepsilon = 1\text{m}$). To get a discrete set of shore points, the shore map

is rasterized into a grid, in which each non-zero cell represents the shore. An arc centered on a pose is drawn with radius d and thickness ε with an angle consistent with the camera intrinsic parameters. For each shore point in the grid, all the poses from which it was seen are identified. An example set of shore points from two different surveys and the points where they overlap are shown in Fig. 2.

The camera poses J that make up the minimal cover set must also satisfy some practical constraints. Poses with an invalid camera configuration or with a high likelihood of motion blur are rejected. Poses from two different surveys without a similar view of the lakeshore are also rejected. In this case, we estimate that two poses have a similar view if their 3D positions are similar and both have similar intersections of the camera axis with the shore at the distance, d , from the boat. The distance between the camera angles is expressed in this way to keep comparable values with the distance between the 3D positions.

The Set Cover Problem is solved with the greedy algorithm shown in Algorithm 1. The method provided the results illustrated in Fig. 3 in less than 30 seconds (i.e., it's tractable). Out of a survey with 50'000 images, roughly 200 are selected for the cover set of the shore, which means over an order of magnitude fewer full runs of image registration will be performed (compared to naively performing image registration for every image in a down-sampled survey). Note that the set of images does not view the entire shore; only all the shore points seen from both surveys with similar views can be covered (about 80%).

```

Let  $L$  be the list of selected view points, initially empty;
while there are shore points to observe do
    Select the valid shore point  $P$  which is the least observed;
    Let  $V$  be a view point such that
         $V$  observes  $P$  and;
         $V$  observes the largest number of unobserved shore points;
    Remove  $P$  and all shore points observed in  $V$  from the list of points to observe;
    Append  $V$  to  $L$ ;
end
return  $L$ 

```

Algorithm 1: Greedy algorithm for maximizing the coverage of the shore with the minimal number of poses.

4.3 Image Registration

Given two poses viewing approximately the same scene from two different surveys, we next run image registration in a local search of several nearby images, and output the image pair with the best alignment score we find (the registration process for a single image pair is shown in Fig. 1). Image registration is performed using a modified version of the SIFT Flow scene alignment algorithm [16], which is designed for matching images with significant amounts of variation between them. SIFT Flow is named as such because a dense image of SIFT descriptors (see [17]) define the matching pattern (the data terms of an MRF) to be optimized between two images.

The algorithm is similar to optical flow in that each pixel is biased to have a similar flow to nearby pixels (a smoothness criteria), and lower degrees of flow are favored (regularization). For two images of approximately the same scene, the alignment score is minimized when the flow lines up salient structures between them.

Because SIFT Flow’s cost function is designed to align the contents of a scene indiscriminately, we add a bias in favor of aligning the lakeshore rather than the sky or the water. A bias to decrease the influence of the sky and the water helps reduce the noise they add to the image alignment process. The sky and the water may compose a majority of each image to be aligned. Yet, they retain little consistent salient structure between surveys. Salient structure can appear in the water if it is reflective, which reduces the likelihood of a good alignment because the reflectivity of the water changes between surveys. The varied appearance of the sky also affects the alignment.

The bias to SIFT Flow’s cost function is derived from the 3D information from visual SLAM. The location of the sky and the water in each image is approximately determined using the estimated 3D locations of tracked landmarks. Although points are mostly only tracked along the shore because most corner features occur there, some are occasionally identified in the sky and the water, which this process essentially filters out. Points with a negative elevation indicate a feature corresponds to the water. Points far away indicate a feature corresponds to the sky. The rest of the reprojected points are interpreted as part of the lakeshore. Given an image and the valid 3D landmarks from visual SLAM, an image mask is created by drawing the reprojected points on an image as a circle (an empirically determined radius, $r=28$, gave the best performance). For each pixel in the non-zero regions, the data terms of SIFT Flow’s objective function are biased (by a factor of 1.5) in favor of aligning the contents there compared to the other areas of the image.

Because belief propagation is used to find the best alignment, which can require a significant amount of computation time to converge in large graphical models, SIFT Flow uses image pyramids to speed up the process. An image pyramid progressively halves the size of the two images for several layers (four in this paper). The search for the best alignment proceeds in a search backwards down the image pyramid, with the flow from each layer bootstrapping the optimization at the next higher resolution. A search window defines the area to be considered for each pixel, and reduces in size with each successive layer.

The final output alignment is chosen after a local search around the two candidate poses to find the two images that align best. SIFT Flow seldom finds a dense correspondence between the first two coarsely aligned images we give it. The perspective difference and the optimization error between the two images is often different enough that an incorrect, high score alignment is found. A better, low score alignment is usually possible between nearby images, which have a slightly different perspective. Therefore, the local search for the best dense correspondence is performed on images at $0, \pm 1.5$, and ± 3.0 second offsets from the two image candidates, for a total of 25 different alignments. To speed up this search, the alignment is only performed for images at the highest level of the image pyramid.

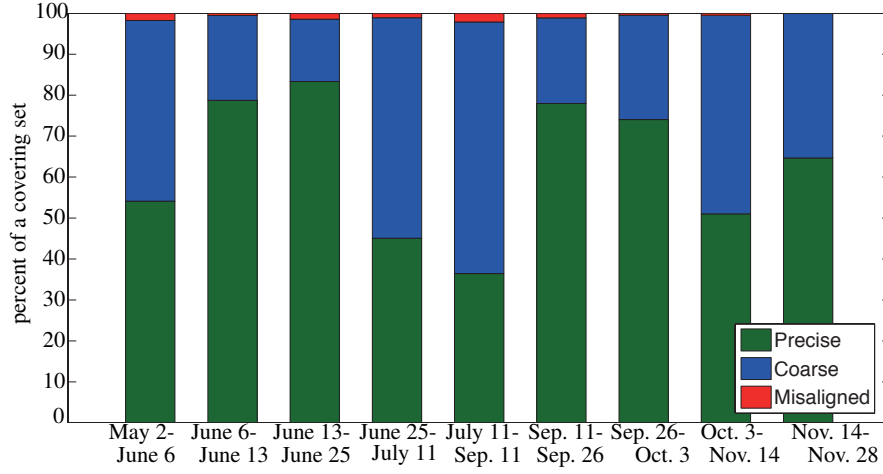


Fig. 4 Alignment quality for comparisons of 10 different surveys. All 10 were performed in 2014.

4.4 Survey Comparison

Although we endeavor to create a system for fully autonomous lakeshore monitoring, including detecting changes autonomously, in this work change detection is left to an end user. Our user interface is designed to exploit human skill at detecting changes in flickering images of a scene. If an image pair from two different surveys is aligned at the pixel level, changes flash on and off when the images are flickered back and forth. If the precise alignment is not possible, a user can always revert to a side-by-side comparison of images. This approach enables a human to perform fast change detection (often requiring only a single flicker) for a survey comparison of a large spatial environment consisting of hundreds of images.

5 Evaluation

5.1 General Alignment Quality

We first evaluate how well our framework aligns 10 consecutive lakeshore surveys, which provides a point of reference for our system’s performance. The 10 surveys are compared in consecutive order for a total of nine different comparisons. The surveys span a total time of 210 days, with seven days the shortest interval between compared surveys and 62 days the longest. For each survey, each image from its cover set and the aligned image from the following survey were flickered back-and-forth in a display. A human evaluated the alignment quality according to three criteria: **precise**) almost the entire image is aligned well with little noise; **coarse**) the images correspond to the same scene and some objects may be precisely aligned; and **misaligned**) the images correspond to different scenes or it is hard to tell they come from the same scene.

The results are shown in Fig. 4. The framework in this paper significantly outperforms that of our previous work in [7], which compared surveys from June 13 and June 25 and achieved 52% precise alignments, 36% coarse alignments, and 12% misalignments. In all the comparisons a significant number of precise alignments are found, although some have more than others. The two cases with the fewest precise alignments involve a comparison with the July 11 survey, which had a much higher water level. The upper half of many images in these two comparisons were precisely aligned, yet because the perspective significantly changed, and the shore-line appeared very different between surveys, SIFT Flow inaccurately extended the shore downward to try to compensate for the large differences in appearance. In the other comparisons, fewer precise alignments are due to sun glare and larger intervals of time between surveys (increasing e.g., the seasonal variation of plants). For every case, however, the few number of misalignments indicates an end user is almost always shown images of the same scene. Thus, because the approach can find good alignments, we next demonstrate its use for change detection.

5.2 *Detected Changes*

While labeling the alignment quality of each comparison, we also saved notable changes between surveys to show our approach is useful for change detection. Six interesting examples are shown in Fig. 5. Five were found in precisely aligned images; the removed treetop was identified in coarsely aligned images. Although the difference between precise and coarse alignments is hard to spot in the figure, it is readily apparent in a flickering display. This is also true for the detection of the cut branch, which is nearly impossible to notice unless the images are precisely aligned and flickered back and forth. Except for the case with people, none of the changes were known of before this analysis. In fact, although we noticed a tree fell in the water after some heavy rain (its branches are sticking out of the water in the Sky and Water example of Fig. 6), we did not know where it came from. Because being able to spot changes depends on image alignment quality, the next section evaluates the robustness of our framework to the variation of appearance across surveys.

5.3 *Robustness to Different Sources of Variation*

Our framework can find many precise alignments in all the surveys only because it is robust to many different, combined sources of variation of appearance. Before two images are precisely aligned the appearance variation between them is often ‘extreme’. Six prototypical examples of robustness to a particular source of variation are shown in Fig. 6. Perhaps the example with the most extreme amount of variation is the one labeled ‘seasonal’. In addition to the foliage depletion captured in this image pair, there is also different illumination, sky, water, shadows, and a globe reflection. Maybe a precise alignment would not have been possible if there was also sun glare. However, there are many cases in which precise alignments are not found. The next section identifies common types of alignment errors.

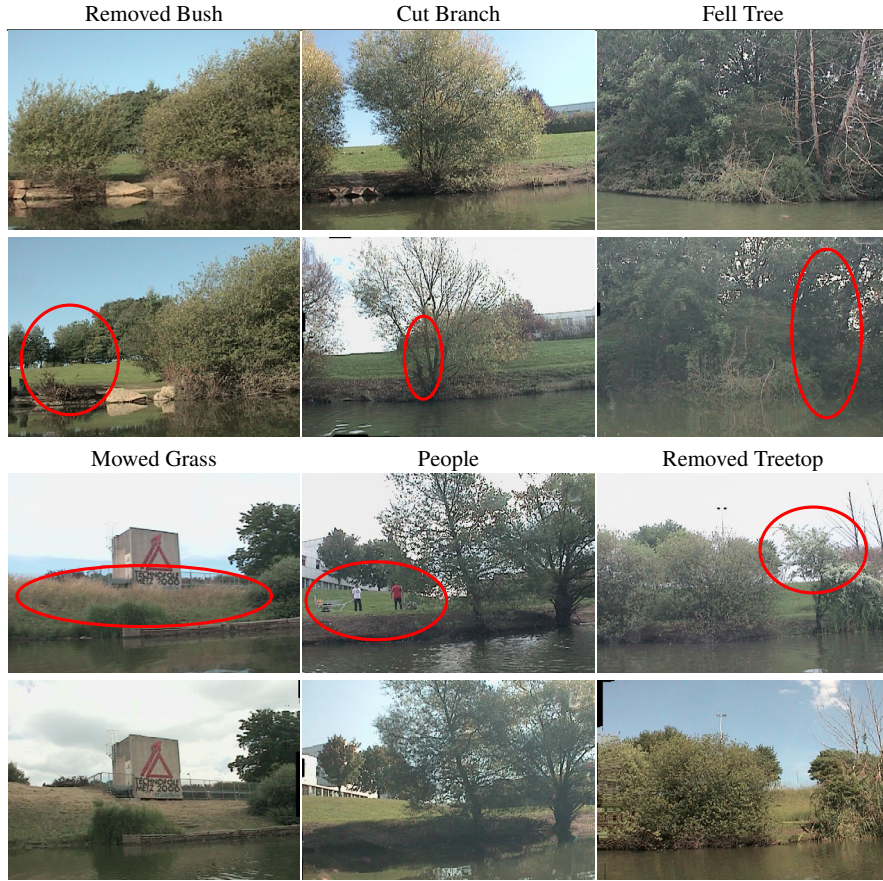


Fig. 5 Six notable changes a human found while comparing the 10 different lakeshore surveys.

5.4 Alignment Errors

In some cases the alignment process adds significant noise to the images, which requires reverting to the unregistered image pair for performing a comparison. Six common ways the precise alignments failed are shown in Fig. 7. Image alignment does not comply with the physics of structures in each warped image, which is apparent in all the cases (and is an effect observed in other image processing work as well, e.g., texture synthesis [15]). Because each pixel is potentially warped differently than nearby pixels, the warp may be inconsistent across the image. Additionally, SIFT Flow may try to align to noise (e.g., sun glare) and changes (e.g., a high-water level water), obfuscating the scene. Notwithstanding errors, most alignments are labeled ‘coarse’ because they are translated versions of the same scenes.



Fig. 6 Precisely aligned image pairs for six different sources of noise, which indicates our approach can be robust to ‘extreme’ appearance variation.

6 Conclusion and Future Work

This paper presented a framework for spatially and temporally scalable lakeshore monitoring. Our approach is based on exploiting scene geometry, using visual SLAM and SIFT Flow, to overcome the variation in appearance of natural environments, and achieve pixel-level data association. Extending prior work, this paper 1) identified a covering set of poses; 2) searched for the best alignment around each candidate pose; and 3) used the lakeshore’s 3D structure to bias the image registration process. These techniques increased survey alignment accuracy with fewer expensive image alignments. This enabled an analysis of ten surveys, in which a human readily identified several changes. The number of precise alignments we obtained amidst ‘extreme’ appearance variation validates our approach.

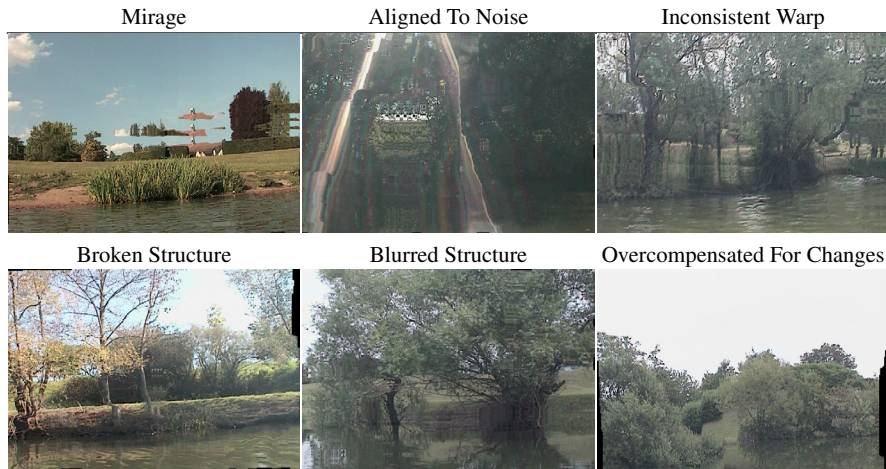


Fig. 7 Six different alignment errors made during image registration.

In future work we plan to further improve our method’s robustness to the variation in appearance between surveys. The many coarsely aligned image pairs are in reach of becoming precisely aligned. One direction is to transition from aligning mostly visual features to placing more weight on aligning the 3D structure of the lakeshore. Another direction is to remove noise (particularly sun glare) before alignment. With these extensions, precise alignments may become even more likely.

Acknowledgements

Funding for this project is provided by the Lorraine Region, France.

References

- [1] Chris Beall and Frank Dellaert. Appearance-based localization across seasons in a Metric Map. In *6th PPNIV*, Chicago, USA, September 2014.
- [2] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *IJRR*, 32(14):1645–1661, 2013.
- [3] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.
- [4] Peter Corke, Rohan Paul, Winston Churchill, and Paul Newman. Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localization. In *IROS*, pages 2085–2092. IEEE, 2013.
- [5] Frank Dellaert. Factor Graphs and GTSAM: A Hands-on Introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012. URL <https://research.cc.gatech.edu/borg/sites/edu.borg/files/downloads/gtsam.pdf>.
- [6] Shane Griffith, Paul Drews, and Cédric Pradalier. Towards autonomous lakeshore monitoring. In *ISER*, 2014.

- [7] Shane Griffith, Frank Dellaert, and Cédric Pradalier. Robot-Enabled Lakeshore Monitoring Using Visual SLAM and SIFT Flow. In *RSS Workshop on Multi-View Geometry in Robotics*, 2015.
- [8] Jinwei Gu, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. *ACM Transactions on Graphics (TOG)*, 28(5):144, 2009.
- [9] Xuming He, Richard Zemel, and Volodymyr Mnih. Topological map learning from outdoor image sequences. *JFR*, 23(11-12):1091–1104, 2006.
- [10] Hordur Heidarsson and Gaurav Sukhatme. Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In *IROS*, pages 3160–3165. IEEE, 2011.
- [11] Gregory Hitz, François Pomerleau, Francis Colas, and Roland Siegwart. State estimation for shore monitoring using an autonomous surface vessel. In *ISER*, 2014.
- [12] Sezal Jain, Stephen T. Nuske, Andrew D Chambers, Luke Yoder, Hugh Cover, Lyle J. Chamberlain, Sebastian Scherer, and Sanjiv Singh. Autonomous river exploration. In *FSR*, December 2013.
- [13] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *IJRR*, 31(2):216–235, 2012.
- [14] Jana Košečka. Detecting changes in images of street scenes. In *Computer Vision—ACCV 2012*, volume 7727 of *LNCS*, pages 590–601. Springer, 2013.
- [15] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graph-cut textures: Image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.
- [16] Ce Liu, Jenny Yuen, and Antonio Torralba. SIFT Flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.
- [17] David Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [18] Colin McManus, Ben Upcroft, and Paul Newman. Scene signatures: Localized and point-less features for localization. In *RSS*, Berkeley, USA, July 2014.
- [19] Peter Nelson, Winston Churchill, Ingmar Posner, and Paul Newman. From Dusk till Dawn: Localisation at Night using Artificial Light Sources. In *ICRA*, 2015.
- [20] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *RAS*, 2014.
- [21] John Rogers and Henrik Christensen. Normalized graph cuts for visual slam. In *IROS*, pages 918–923. IEEE, 2009.
- [22] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *IJRR*, 29(8):958–980, 2010.
- [23] Anbumani Subramanian, Xiaojin Gong, Jamie Riggins, Daniel Stilwell, and Christopher Wyatt. Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications. In *OCEANS*, pages 1–6. IEEE, 2006.