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

Shane Griffith and Cédric Pradalier

Abstract This paper presents an application of autonomous surface vessels for long-term observation of lakeshore environments. This paper specifically addresses challenges of data association in the midst of 'extreme' variation of appearance. Our domain consists of 55 surveys of a 1km lakeshore collected over a year and a half. In previous work our framework aligned images between different surveys using visual SLAM and dense image correspondence with SIFT Flow. This paper shows how the visual coverage of a lakeshore can be maximized with a minimized number of images, addressing the Set Cover Problem. We also improve the pixel-level image alignment of SIFT Flow using the 3D landmark positions from visual SLAM to bias the dense correspondence. Compared to previous work our method is significantly faster and finds significantly more precise alignments. The large number of precise alignments demonstrate robustness to variation in appearance of the sky, the water, changes in objects on a lakeshore, and the seasonal changes of plants. We also show our framework enables a human to detect changes between surveys that would otherwise go unnoticed.

1 Introduction

This paper describes the use of an autonomous surface vessel for long-term lakeshore monitoring, and provides a foundational image processing framework to assist in the visual detection of changes in consecutive surveys of a natural environment. Currently, planes and satellites are a go-to platform for surveys of lakeshores and many other natural environments because they can capture large swaths of an area in in-

Shane Griffith
GeorgiaTech Lorraine e-mail: name@email.address

Cédric Pradalier
GeorgiaTech Lorraine e-mail: name@email.address

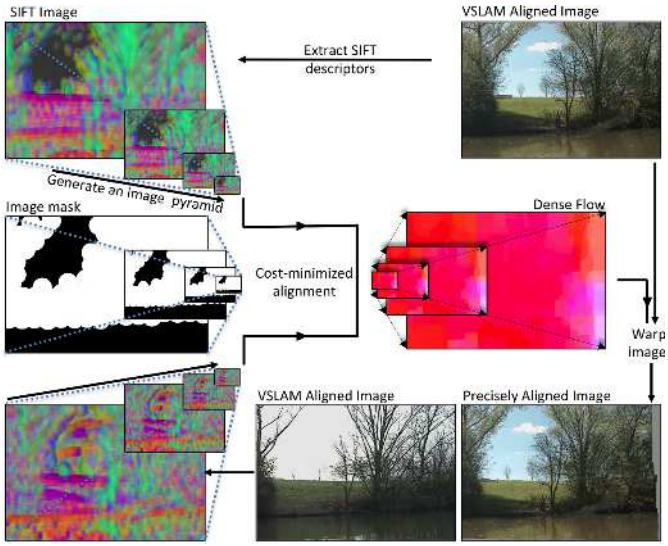

Fig. 1 Our lakeshore monitoring framework can provide a user with aligned views of a lakeshore from multiple surveys, which enables quick change detection.

creasingly higher resolution images. Yet, mobile robots may be more suited to many monitoring tasks because the deteriorating quality of a lake is most noticeable in the field, their view is unobstructed by tree cover, and they can autonomously provide a high-resolution view of an entire shoreline.

This paper extends work submitted as [6], which introduced methodology for achieving high resolution, pixel-level alignment between fortnightly surveys of a lakeshore. Our framework uses visual SLAM (see e.g., [12, 2, 7, 1]) to identify coarsely aligned images from different surveys and then it applies SIFT Flow [14] to precisely align them. Because SIFT flow captures the global structure of a scene using whole images of feature descriptors, in contrast with approaches based on sparse local feature matches, it can often robustly find whole-image correspondences across the variation in appearance typical of many natural environments. Once images are precisely aligned, a human inspecting them can easily spot if something changed. At this stage, given the difficulty of visually understanding natural scenes, we are still assuming that the inspection task will be left to a human, but we endeavor to make his/her task as efficient as possible. In comparison with our previous work, this paper makes the following contributions: we show how to make use of the information from the 3D structure recovered by the Visual SLAM layer in the dense correspondence, we discuss how to select a minimal set of shore views to cover the complete environment, and we provide more exhaustive results covering a wider temporal span of the data we collected.

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 for autonomous lakeshore monitoring. In spite of the large variation of appearance across surveys, our frame-

work provides a human with aligned images and a way to quickly detect changes between them. The large number of precise alignments demonstrate robustness to variation in appearance of the sky, the water, changes in objects on a lakeshore, and the seasonal changes of plants. We also show our framework enables a human to detect changes between surveys that would otherwise go unnoticed.

2 Related Work

The field of Simultaneous Localization and Mapping (SLAM) provides a foundation for localizing a robot and mapping monitored spaces. Utilizing SLAM in natural environments requires a system made to handle the large spatial scale. This paper uses iSAM2 [11], which restricts optimization to the subset of variables affected by new measurements.

Performing Visual SLAM in any outdoor environment adds two additional constraints: the systems needs to be able to handle an environment which is not completely rigid and that presents a very high level of self similarity. For instance, the leaves of a willow tree are all very similar and not so different from another lakeshore tree. The variation of appearance over time increases the difficulty of finding associations between images of an outdoor environment. Some approaches rely on point–based features such as SIFT (e.g., [12, 2, 7]) for performing data association.

Because point–based feature matching is often not robust to common sources of variation in outdoor appearance, some work has focused on directly using, or modifying, whole or parts of images. Neubert et al. [17] 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. [16] 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 [3].

Being able to operate in a lake and map the location of a lakeshore is an essential task of lakeshore monitoring, which some papers have already started to address. Heidarsson and Sukhatme [8] and Subramanian et al. [18] map a lakeshore and the locations of obstacles from the visual perspectives of their ASVs. Jain et al. [10] 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. [9] 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.

3 Experimental Setup

We used Clearpath's Kingfisher autonomous surface vessel (ASV) for our experiments. It is about 1 meter long and 2/3 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, which can turn it by differential steering. 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 outfitted with several sensors befitting an autonomous surface vehicle. A prominent 704x480 color pan-tilt camera stands on top, capturing images at 10 frames per second. Beneath it sits a single scan line laser-range finder 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 continually deployed the robot up to once per week since August 18, 2013. This paper analyzes data from ten different surveys, which cut across 7 months of variation. Each survey was performed in the daytime on a weekday in sunny or cloudy weather, at various times of the day. All ten consisted of one complete run around the entire lakeshore, including the island.

4 Methodology

Our long-term lakeshore monitoring framework aligns images using a coarse-to-fine process. It consists of 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 as much of the lakeshore as possible. Third, given two poses facing the same scene from two different surveys, a local search 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 $\mathscr{A}^k=\{\mathscr{T}_i^k, \mathscr{I}_i^k, \hat{C}_i^k, \omega_i^k\}_{i=1}^{|\mathscr{A}^k|}$ every tenth of a second, where $\mathscr{T}$ is the current time, $\mathscr{I}$ is the image from the pan-tilt camera, $\hat{C} \in$ SE(3) is the estimated camera pose, and ω is the boat's angular velocity as measured from its IMU. The estimated camera pose is derived from the boat's GPS position, the measured heading from the compass, and the pan and tilt positions of the camera. 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 use visual SLAM to improve our estimates of the camera positions.

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]. The detailed process is described in [6].

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 there is a lot of redundancy in the images, 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 maximizes the coverage of the lakeshore.

Another name for this is the "Set Cover Problem" (SCP) [4]. Adapted to our case, the SCP can be expressed as follows. Let $\mathscr{S}$ be the set of all the observable positions of the shore of a lake in an entire survey. Each image, i, of the survey observes a subset $\mathscr{I}_i$ of these shore points, where $\mathscr{S} = \bigcup_{i \in I} \mathscr{I}_i$. The goal is to find

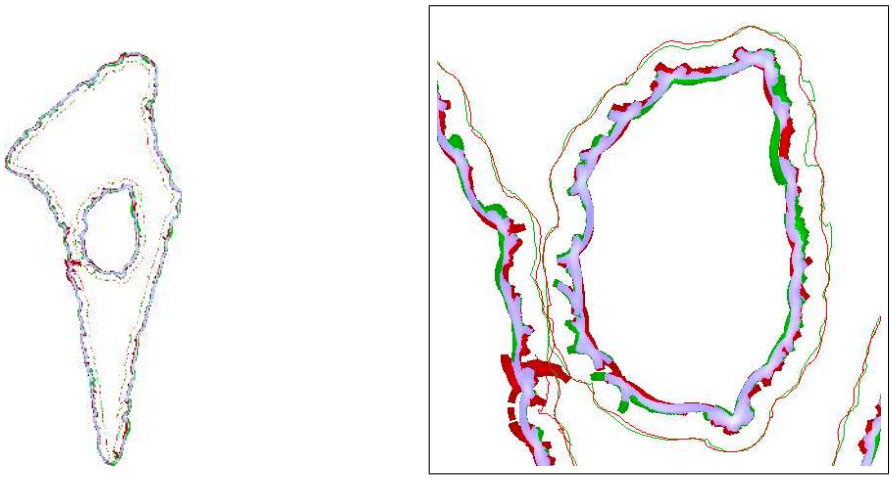

Fig. 2 The recorded trajectory of the boat and the shore points it sees for two surveys. The shore points seen from the red trajectory are displayed in red, those seen from the green trajectory are green, and those seen from both are blue. The right image shows a closeup version of the left one.

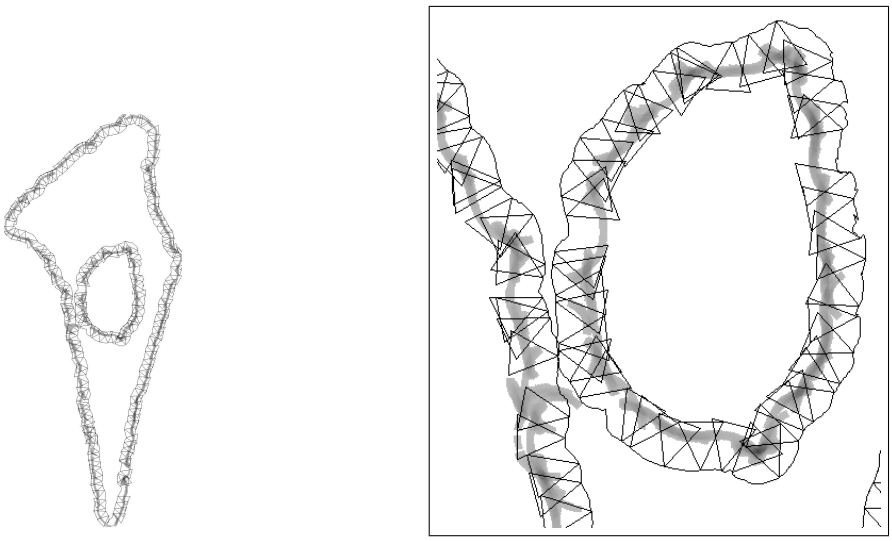

Fig. 3 The cover set for the red survey from Fig. 2, which accounts for co-visibility with the green survey. Black triangles indicate the visibility frustum of the selected images. The right image is a zoomed-in version of the left one.

a set of images J for which $\mathscr{S} = \bigcup_{j \in J} \mathscr{S}_j$ 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.

Therefore, to solve our problem, the set of shore points that compose $\mathscr{S}$ is identified, candidate poses that do not satisfy some basic constraints are rejected, and a greedy algorithm is run to find a minimized set of poses that cover as much of the lakeshore as possible.

The set of shore points that compose $\mathscr{S}$ is identified using the optimized boat poses from visual SLAM. Because the robot is controlled to move at a constant distance d (10m in our case) from the shore, every point at distance $d \pm \varepsilon$ (where $\varepsilon = 1m$) in the camera frustum is considered a shore point. This is implemented by rasterizing the shore map into a pixel map and drawing a thick 10m arc centered on every pose with an angle consistent with the camera intrinsic parameters. Every pixel on the map represents the shore. For each pixel on the map, 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.

For practical reasons, the minimal covering set can only consist of poses that satisfy some additional constraints. Poses with an invalid camera configuration or with a high likelihood of motion blur are rejected. Poses without a similar view of the lakeshore in a compared survey are also rejected. In this case, a pair of poses from two different surveys are considered useable if their 3D positions are similar and both have similar intersections of the camera axis with the shore at a specified 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. Out of a survey with 50'000 images, roughly 200 are selected for the cover set of the shore.

```
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. Image registration is performed using a modified version of the SIFT Flow scene alignment algorithm [14], 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 [15]) define the matching pattern 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 shifts each pixel in a way that salient structures line up.

Because SIFT Flow uses belief propagation in a high dimensional space to find the best alignment, which can require a significant amount of computation time, image pyramids are used 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.

We modify the objective function of SIFT Flow using the 3D landmark locations from visual SLAM for an improved image correspondence. Salient structure can appear in the water if it is reflective, which reduces the likelihood of a good alignment. A cloudy sky may also affect the alignment. Landmark positions from visual SLAM too far away or with negative height indicate sky or water. Because many feature tracks are found in each image in the optical flow step of visual SLAM, we can approximately determine where the lakeshore is in each image. Given an image and the optimized 3D points, we create an image mask by drawing the reprojected points on an image as a circle with radius 28. For each pixel in the mask, the objective function of SIFT Flow is biased (by a factor of 1.5) to align its contents compared to the unmasked regions.

The final output alignment is chosen after a local search for the best aligned images around the two candidate poses. 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 $\pm$ 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.

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 spotting 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

Our first experiment tests how well our framework can align lakeshore surveys, in general, across large spans of time and in the midst of significant environment variation. We chose ten different surveys for this analysis, which are compared in consecutive order for a total of nine different comparisons. The surveys span a total time of 210 days, with 7 days the shortest interval between compared surveys and 62 days the longest. For each survey, each image from its covering 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 [6], 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 least 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 shoreline 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. Therefore, because the approach can find good alignments, we next showed its use for change detection.

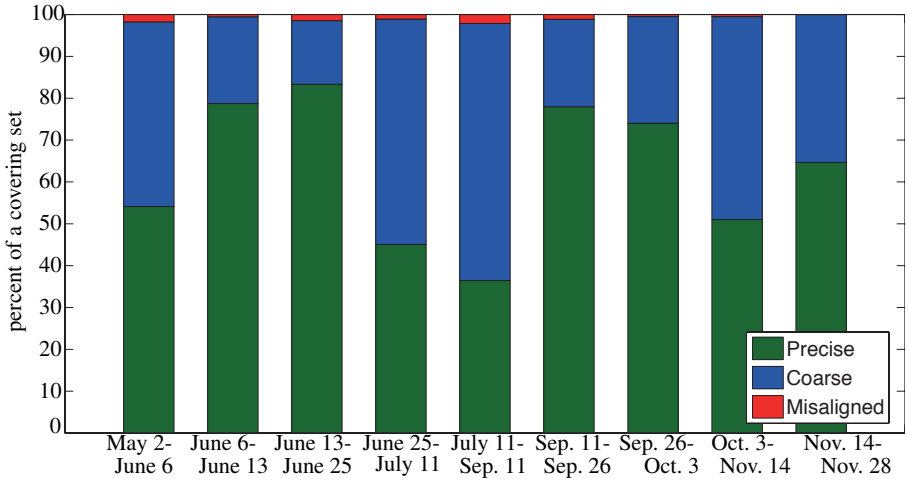

Fig. 4 Alignment quality of ten different survey comparisons. All ten surveys were performed in 2014.

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 of the changes were spotted in precisely aligned images; the removed treetop was spotted in coarsely aligned images. Except for the case with people, none of them were known of before the survey comparison. In fact, although we noticed a tree fell in the water after some heavy rain (it's branches are sticking out of the water in the Sky and Water example of Fig. 6), we did not know where it came from.

Changes are easier to spot in precisely aligned images. For example, the cut branch is nearly impossible to notice unless the images are precisely aligned and flickered back and forth. In coarsely aligned images it may be mistaken for a perspective difference, if noticed at all. In contrast to coarse images, in precisely aligned images small changes are as readily spotted as large changes. Yet, finding large changes can still require a long search if images are only coarsely aligned. Of course, detecting any changes becomes easier as images become more precisely aligned, which is, in turn, correlated with the robustness of our approach to the variation of appearance across surveys, which is described in the next section.

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

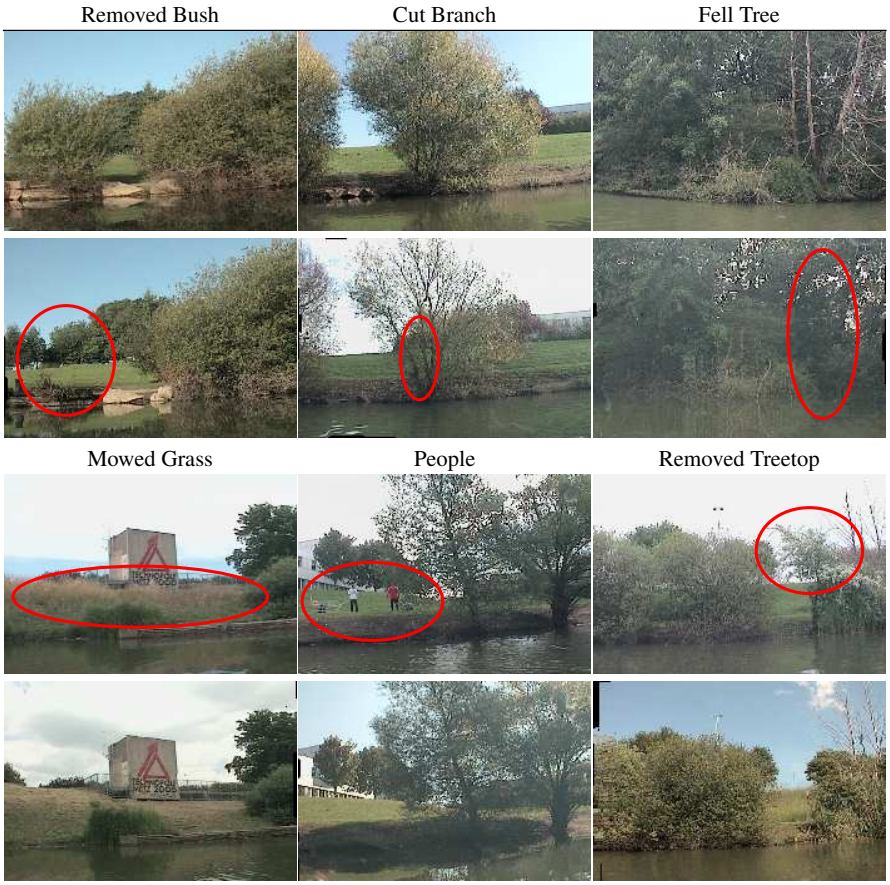

Fig. 5 Six examples of changes found by a human while comparing different surveys of a lakeshore.

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. There are, indeed, many cases like these in which precise alignments are not found, however, which we describe in the next section.

Fig. 6 Six different sources of noise and precisely aligned image pairs, which show that our approach is robust to 'extreme' sources of appearance variation.

5.4 Alignment Errors

In some cases the alignment process adds significant noise to the images, which requires reverting to the coarse image alignment 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 [13]). Because each pixel is potentially warped differently than nearby pixels, the warp may be inconsistent across the image. Additionally, SIFT Flow can try to align to noise (e.g., sun glare) and changes (e.g., a high-water level water), creating a blurred structure. Notwithstanding errors, most alignments are labeled 'coarse' because they are translated versions of the same scenes.

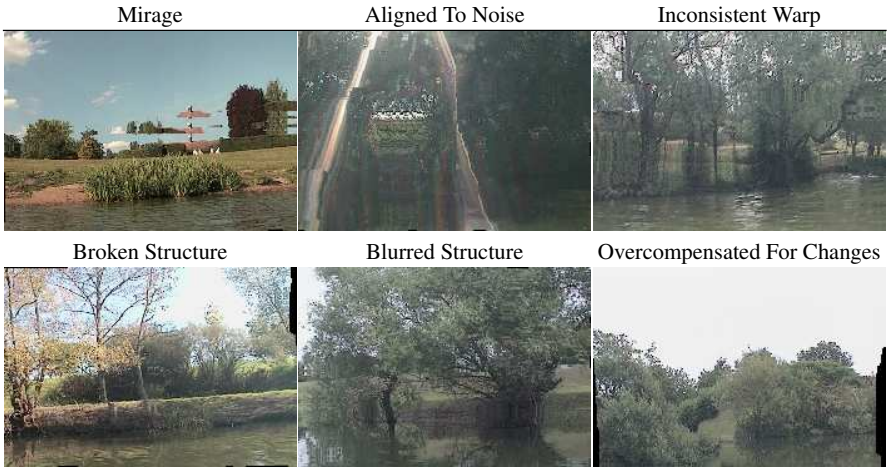

Fig. 7 Six different alignment errors made during image registration.

6 Conclusion and Future Work

This paper presented a spatially and temporally scalable approach to long-term lakeshore monitoring. It significantly extended our previous work based on aligning surveys using visual SLAM and SIFT Flow. Finding a covering set of poses for each survey reduced the number of computationally expensive image alignments, which enabled a tight, local search around each candidate pose for the most precise alignment. The use of the lakeshore's 3D structure in the image registration process increased the likelihood of finding precise alignments. Together, these enhancements help make our approach robust to the 'extreme' variation in appearance typical of lakeshore environments.

In future work we plan to improve upon our method's robustness to the variation of appearance between surveys. There are still many coarsely aligned image pairs, which are in reach of becoming precisely aligned. One direction is to transition from a process based mostly on aligning visual features to one that also places significant weight on aligning the 3D structure of the lakeshore. Another direction is to add an image pre-processing step to remove noise before alignment. With these extensions, finding precise alignments may become even more likely.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building Rome in a Day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Chris Beall and Frank Dellaert. Appearance-based localization across seasons in a Metric Map. In *6th PPNIV*, Chicago, USA, September 2014.

[3] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *IJRR*, 32(14):1645–1661, 2013.

[4] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.

[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, Frank Dellaert, and Cédric Pradalier. Robot-Enabled Lakeshore Monitoring Using Visual SLAM and SIFT Flow. In *submitted to Robotics Science and Systems (RSS)*, 2015.

[7] Xuming He, Richard S Zemel, and Volodymyr Mnih. Topological map learning from outdoor image sequences. *JFR*, 23(11-12):1091–1104, 2006.

[8] Hordur Heidarsson and G Sukhatme. Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In *IROS*, pages 3160–3165. IEEE, 2011.

[9] Gregory Hitz, François Pomerleau, Francis Colas, and Roland Siegwart. State estimation for shore monitoring using an autonomous surface vessel. In *ISER*, 2014.

[10] 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.

[11] 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.

[12] Jana Košecka. Detecting changes in images of street scenes. In *Computer Vision–ACCV 2012*, volume 7727 of *Lecture Notes in Computer Science*, pages 590–601. Springer, 2013.

[13] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graphcut textures: Image and video synthesis using graph cuts. In *ACM Transactions on Graphics (ToG)*, volume 22, pages 277–286. ACM, 2003.

[14] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 33(5):978–994, 2011.

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] Colin McManus, Ben Upcroft, and Paul Newman. Scene signatures: Localized and point-less features for localization. In *RSS*, Berkeley, USA, July 2014.

[17] Peer Neubert, Niko Sünderhauf, and Peter Protzel. Superpixel-based appearance change prediction for long-term navigation across seasons. *RAS*, 2014.

[18] Anbumani Subramanian, Xiaojin Gong, Jamie N Riggins, Daniel J Stilwell, and Christopher L Wyatt. Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications. In *OCEANS*, pages 1–6. IEEE, 2006.