Robot–Enabled Lakeshore Monitoring Using Visual SLAM and SIFT Flow

Shane Griffith
Georgia Institute of Technology
sgriffith7@gatech.edu

Frank Dellaert
Georgia Institute of Technology
dellaert@gatech.com

Cédric Pradalier
GeorgiaTech Lorraine - CNRS UMI 2958
cedric.pradalier@gatech.edu

***Abstract*—This paper establishes an autonomous monitoring framework to augment a human's ability to detect changes in lakeshore environments. This is a large spatial and temporal scale study, which analyzes data from eight different surveys of a lakeshore collected over 11 months with an autonomous surface vehicle. Despite the variation in appearance across surveys, our framework provides a human with aligned images and a way to readily detect changes between them. First visual SLAM is used to find a coarse alignment of images between surveys, and second, SIFT Flow is applied to achieve dense correspondence. The aligned images are flickered back-and-forth in a user display, which enables a human to rapidly detect changes. Results show our method can align images in the midst of variation in appearance of the sky, the water, changes in objects on a lakeshore, and the seasonal changes of plants.**

I. Introduction

This paper introduces a framework for the data association of images of a natural environment (see Fig. 1), with which an autonomous surface vehicle (ASV) enables lakeshore monitoring. Robots 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. Yet, the data association needed to automatically process surveys remains difficult. Images vary with the time of day, the weather, the seasons, etc. Most related work addresses lighting changes in urban environments and is limited to sparse, feature-based correspondence. Enabling high-resolution, pixel-wise comparisons between surveys requires dense correspondence, which cuts through the variation in appearance of natural environments.

This paper achieves dense correspondence between surveys of a natural environment. Our framework provides a human with aligned images and a way to readily detect changes between them. First, visual SLAM is used to identify images of the same scene from different surveys. Second, SIFT Flow is applied to achieve dense correspondence. A user interface displays the aligned image pairs, which are flickered back-and-forth to enable rapid inspection.

This study analyzes data from eight different surveys of a lakeshore collected over 11 months with an ASV. We capture significant variation in appearance across surveys. Despite this, results here indicate our framework can precisely align a majority of the scenes. Although our setup and experiments survey a lakeshore, this approach is applicable to other types of vehicles and environments with few changes.

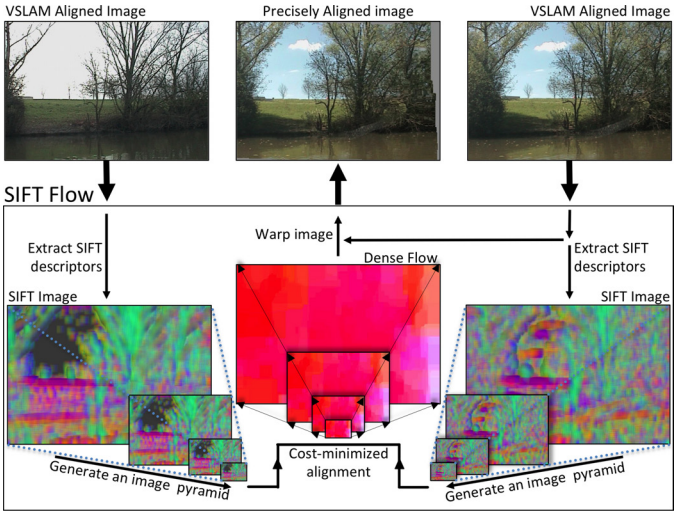

Fig. 1. Our framework for coarse-to-fine alignment of surveys of natural environments, which combines visual SLAM and SIFT Flow.

II. Related Work

The field of Simultaneous Localization and Mapping (SLAM) provides a foundation for localizing a robot and mapping monitored spaces. Large spatial scales of natural environments, however, challenge SLAM systems. Most approaches that address scalability split optimization into smaller problems. For example, Ni et al. [15] divide-and-conquer the SLAM problem using submaps, which are optimized separately and then again during recombination. This paper uses iSAM2 [10], which restricts optimization to the subset of variables affected by new measurements.

Long outdoor deployments also challenge SLAM systems. Data association in an outdoor environment becomes more difficult as its appearance varies over time. Many different approaches have been proposed to address this. Some rely on point–based features for data association [6, 4, 9]. Because point–based feature matching is often not robust to outdoor environment variation, some work has focused on directly using, or modifying, whole or parts of images [14, 2, 13]. This paper aligns whole images using SIFT Flow, which combines the accuracy of point-based feature matching with the robustness of whole–image matching.

Some papers have already started to address working in

a lake and mapping the location of a lakeshore, which is is an essential task of lakeshore monitoring. Heidarsson and Sukhatme [7] map a lakeshore and the locations of obstacles from the visual perspective of their ASV. In case a robot repeatedly visits the same lakeshore, Hitz et al. [8] 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.

We first introduced our work towards autonomously traversing the perimeter of a lakeshore while surveying it in [5]. There we speculated the structure of a scene could serve as the basis for data association. Accordingly, in this paper visual SLAM and SIFT Flow use the structure of a lakeshore to align image sequences from consecutive surveys. A human compares the aligned images, which exploits human skill at detecting changes between flickering images of a scene.

III. Experimental Setup

A. Robot

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, which can turn it by differential steering. The battery lasts about an hour at 0.4 m/s with its current payload.

Our Kingfisher has a pan-tilt camera, a laser rangefinder, a GPS receiver, a compass, and an IMU. The camera captures 704x480 color images at 10 frames per second. The laser rangefinder scans a point just above the water along a 270 degree arc, which provides a distance estimate for objects closer than 20 meters.

B. Environment

We used our ASV to survey Symphony Lake in Metz, France. The lake 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. Fishermen occasionally sat along it.

C. Behavior

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 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. A predefined waypoint along the perimeter marks the transition to surveying the island. At this point the pan-tilt camera is rotated from pointing starboard–side to pointing port–side, and the planner begins optimizing the boat's proximity to land on its port side. The robot can perform an entire survey autonomously; however, we occasionally took control to swap batteries, fix unexpected control failures, or avoid fishing lines and debris.

We deployed the robot up to once per week over a period of 1.5 years, which began August 18, 2013. Each survey was captured during the daytime on a weekday in sunny or cloudy weather. This paper analyzes data from eight different surveys captured in 2014, which span 11 months of variation.

IV. Methodology

Enabling interactive monitoring between surveys consists of a two–step coarse-to-fine alignment process. First, the rough location where each image was captured is identified. Second, a pixel-wise alignment is computed for images of the same scene from two different surveys. These steps consist of Visual SLAM and SIFT Flow, respectively.

A. Data Collection

A survey represents a collection of images, a trajectory of camera poses, and other useful information about the robot's movement. During a survey, k, the robot acquires the tuple $\mathcal{A}^k \cup \{\mathcal{I}_t^k, p_t^k, \omega_t^k\}$ every tenth of a second, where t is the current time, $\mathcal{I}_t^k$ is the image from the pan-tilt camera, $p_t^k \in$ SE(3) is the measured camera pose, and ω_t^k is the boat's angular velocity as measured from its IMU. The camera pose combines the boat's GPS position, the compass heading, and the pan-tilt camera's position.

B. Visual SLAM

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 a visual slam framework to improve our estimates of the camera poses. Landmarks for visual SLAM are obtained with generic feature tracking from OpenCV, which is based on detecting 300 Harris corner features and then tracking them using the pyramidal Lucas–Kanade Optical Flow algorithm as the boat moves (see [5] for details). We use graph–based SLAM to obtain optimized camera poses and landmark positions, which are represented in a factor graph along with the constraints between them (see Fig. 2). The GTSAM framework [3] is used to perform the bundle adjustment step. To reduce data redundancy and speed up computation time, each survey is down–sampled in time by a factor of five for optimization.

A factor graph is constructed over time as a survey is acquired. A node, x_t, is added for the camera pose at each time step, t. A node, l_j, is added for the position of each newly observed landmark. We also add a node, v_t, for the velocity of the robot at each time step, which adds a bit more complexity to the factor graph. The boat's relatively constant velocity provides a kinematic constraint between two consecutive poses. Usually odometry readings provide the kinematic constraints. Because our ASV is, however, propelled by jet thrusters rather than wheels, typical odometry readings are lacking.

Whereas nodes in the factor graph represent the variables to be optimized, factors in the factor graph describe the constraints on each variable. The measured camera pose, p_t,

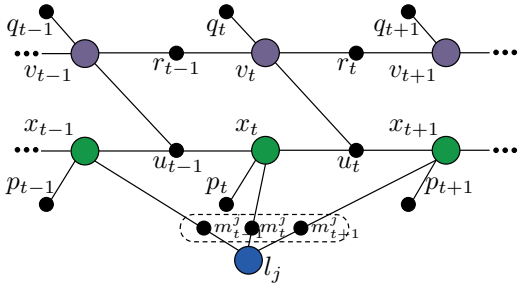

Fig. 2. Factor graph of our SLAM problem. Nodes define landmarks (blue), the robot's pose (green), and the robot's velocity (purple). Factors (black) define constraints on the values of nodes. The dotted line denotes a smart factor, which isolates one landmark's constraints during optimization.

constrains x_t. Landmarks seen at x_t are constrained to project their 3D positions, l_j, to their observed 2D pixel locations, m_t^j. The factor, u_t, describes x_{t+1} as a function of x_t, v_t, and some deviation. Consecutive velocity nodes are related by a factor, r_t, which forces the velocity to change slowly between time steps. A prior, q_t, keeps v_t relatively constant.

The optimized variable assignment, $x_t|_{t=1}^n$, $l_j|_{j=1}^z$, and $v_t|_{t=1}^n$, reduces the error among all the variables according to how tight each constraint is. The tightness of a constraint is its expected standard deviation, which is manually specified. For example, the standard deviation of the compass measurement is roughly 10 degrees due to distortions of the magnetic field during peak motor currents. Because several noisy sensor readings are combined to describe the value of a variable, an optimal variable assignment balances the error among them. The goal of optimization is to find values for all the variables that minimize the total error in the factor graph.

We perform optimization using iSAM2 [10]. iSAM2 converts a factor graph into a Bayes tree to make incremental bundle adjustment computationally feasible. A Bayes tree hierarchically organizes the variables with large amounts of error (the ones measured most recently are usually on top), which prioritizes the optimization. Variables' values are refined over several iterations using the Levenberg–Marquardt algorithm.

To further reduce the computation time of optimization, we also use smart factors [1]. Smart factors employ the Schur complement to split a large optimization problem into smaller, equivalent subproblems. They also eliminate landmarks observed through a degenerate motion or only once. In this paper, the optimization for each landmark, l_j, is isolated with them.

The result of this process is optimized camera poses, $x_t^k|_{t=1}^n$, and landmark positions, $l_j^k|_{j=1}^z$, for each survey, $\mathcal{A}^k$. We use the optimized camera poses to identify images of the same scene from different surveys. We use the optimized landmark positions to compute the average reprojection error for each image, which indicates how closely an image captures its true scene. Using the values for, $l_j^k|_{j=1}^z$, and the initial measurements, $f_j = m_{t_{first}}^j, ..., m_{t_{last}}^j$, the reprojection error was found to average around 3 pixels for all the surveys.

C. Image Registration

Given two different images of roughly the same scene, we aim to facilitate their comparison using image registration. Finding a dense correspondence is, however, challenging due to the dramatic variation in appearance between the images. The structure of the scene may be their most consistent feature [5]. Accordingly, we align images using SIFT Flow [11], which is designed to register similar structures even if the images capture different scenes.

The first step of image registration is to determine which two images to align. Here we use the image $\mathcal{I}_j^{k-1}$ in survey $k-1$ that most likely captures the same scene as image $\mathcal{I}_i^k$ in survey k. They capture the same scene if the camera was at a similar location and pointed at the same thing. Thus, the L_2 distance is computed for both the camera position and the position 10m in front of the camera.

SIFT Flow starts by constructing a 'SIFT image' of each image, which defines the data to be aligned. A SIFT image has the same height and width as the original image, but with 128 channels. Each pixel in the original image is essentially replaced by a 128-byte SIFT descriptor (see [12]). Local gradient information from the 16x16 pixel neighborhood of a pixel is captured in the feature. Two pixels match if the L_1 distance between their descriptors is low (i.e., less than 300).

A dense flow, w, specifies how to align one SIFT image, S_j^{k-1}, to the other, S_i^k, and is found by minimizing a cost function over candidate flows. For pixel p in S_j^{k-1}, the candidate flow $w_p = \{u_p, v_p\}$ determines what pixel it aligns with in S_i^k, where $u_p, v_p \in [-h..h]$, and h is the search window parameter. The cost of w sums the SIFT descriptor distances, regularization terms, and smoothness terms:

$$\sum_p min(|S_j^{k-1}(p) - S_i^k(p + w_p)|_1, t) \quad (1)$$
$$+ \sum_p \nu|u_p + v_p|$$
$$+ \sum_{q \ adj. \ to \ p} min(\alpha|u_p - u_q|, d) + min(\alpha|v_p - v_q|, d)$$

The empirical parameters ν=0.005*255, α=2.5*255, and d=40*255 define the weight of each term and were held constant. The value of t is the median of all the descriptor distances computed between the two images. The best flow, $\overset{*}{w}$, dictates that descriptors of aligned pixels be similar, smaller flows be preferred, and adjacent pixels have similar flows.

The search for the best flow, $\overset{*}{w}$, progresses through image pyramids, which reduces the search time. An image pyramid consists of recursively halved images (four in this paper). The best alignment identified at the low-resolution image pair is refined through increasingly higher resolutions. From low- to high-resolution, $h = 5$, 3, 2, and 1, respectively.

The result of this process is a flow, $\widehat{w}$, which specifies how to interpolate $\mathcal{I}_j^{k-1}$ to align with $\mathcal{I}_i^k$, and an alignment score, ∇_j^k. The score is only computed from the first term in Eq. 1 (the SIFT descriptor distance). We repeat this process for every fifth image in $\mathcal{I}^k$ to enable survey comparison.

Fig. 3. Aligned image pairs for six consecutive places along the lakeshore from the June 25, 2014 and the June 13, 2014 surveys. The mowed grass is one of the more salient changes between them, which appears in all six.

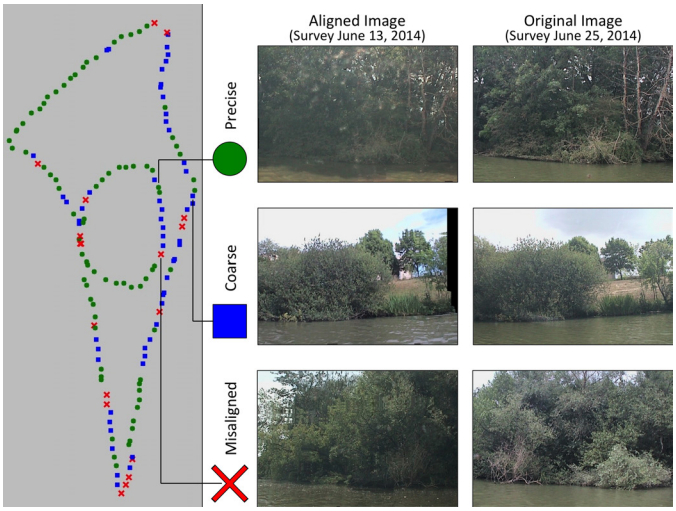

Fig. 4. The quality of the aligned images between the June 13, 2014 and the June 25, 2014 surveys around the entire perimeter of the lake and the island. Representative examples of each alignment class are shown.

D. Survey Comparison

Our user interface enables an end user to rapidly inspect surveys. A human is shown a subset of the aligned images to reduce redundant comparisons. New sections of the shore start about every 9.27m due to the camera's focal length and its 10m distance from the shore. Image pairs selected for display have a low reprojection error after optimization (less than 8 pixels on average), a low alignment score after SIFT Flow (less than 1710 per pixel), and advance the survey comparison up the shore the most (up to 9.27m). If these thresholds are not satisfied, only the alignment score is used. Otherwise the selected image pair is the one of the next section of the lakeshore, 9.27m away.

Each image pair is flickered back-and-forth in the display to draw attention to changes. Although only a subset of all the aligned image pairs are selected for display, the time spent inspecting every scene can add up for large lakes in a side-by-side comparison. Fortunately, humans are highly sensitive to changes in images of a scene if the images are flickered back-and-forth [16]. Therefore, in our display a user scrolls through the selected images from a survey while each image and the one aligned to it flicker back-and-forth. In case two images align poorly, the user can toggle the side-by-side display of the non-warped images.

V. Experiments

A. Alignment Quality Between Two Surveys

The first experiment evaluates how well our framework aligns two surveys, which consists of hand–labeling the quality of their aligned image pairs. We assessed the alignment quality for the surveys from June 13 and June 25. The quality is divided into three classes: *precise*, *coarse*, and *misaligned*. **(precise)** Ideally, each image pair would closely align, or at least be similar enough to quickly spot any changes between them. **(coarse)** However, perspective differences may be difficult to compensate for. Additionally, warping an image can introduce artifacts, which obscure changes. **(misaligned)** In the worst case, unrecognizable images or images of different scenes may be selected for alignment.

The start of the comparison is shown in Fig. 3 with archetypical examples from each class and the quantitative analysis in Fig. 4. The comparison for the entire lakeshore and the island involved 155 different image pairs. Of those, 81 were precisely aligned. Change detection was still possible in an additional 56. A correspondence was not immediately discernible in the remaining 18.

Changes were easily discernible as the most salient thing in precisely aligned images because they flickered on and off. The rest of the scene was mostly static. The major structures (tree trunks, buildings, large branches) usually entirely overlapped. Minor features (e.g., leaves and twigs) sometimes slightly shifted or changed color. In some cases, motion parallax was observed in a few objects due to perspective differences; minor noise cases were counted as precise.

The side-by-side display was useful for identifying changes in coarsely aligned images. Images initially too far apart, with too few salient features, or with artifacts due to SIFT Flow were not aligned well and appeared noisy in the flickering display. Extra time was required to rule out the different sources of noise.

The misaligned images were difficult to compare in either display. They occurred at curves and transitions where the camera's position significantly varied. For example, in one

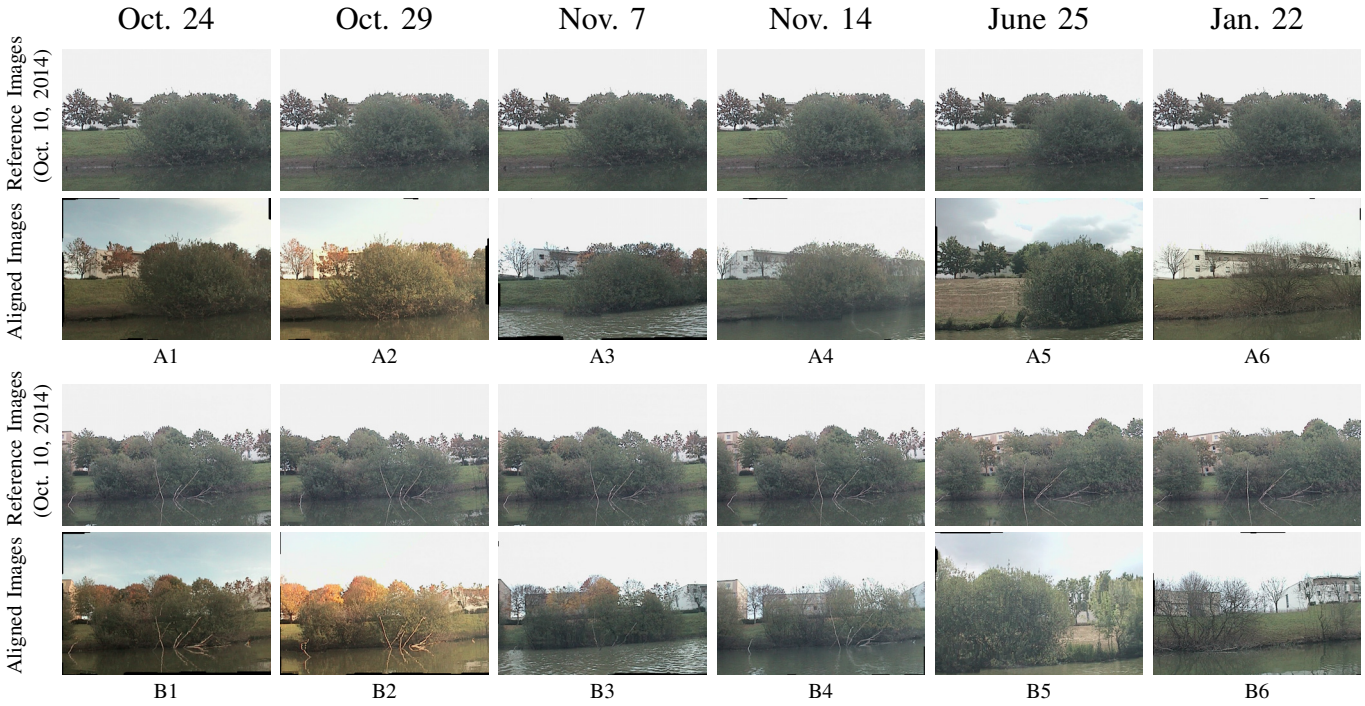

Fig. 5. Surveys from six different dates are aligned to the survey from Oct. 10, 2014 at two different sections of the lakeshore, denoted A1-A6 and B1-B6. Each warped image is shown with the reference image it aligned to, which is slightly different for each comparison.

survey the pan-tilt camera switched from facing the island to facing the lakeshore while in the other survey it still faced the island. Different scenes were matched near the south end of the lake. In other cases, images only slightly overlapped.

B. Robustness to Time Between Surveys

Because models of the environment have limited lifespan and lose applicability as time passes, the second experiment tests the suitability of our approach for infrequent surveys across seasons. We evaluate the quality of an alignment at two different sections of the lakeshore for increasingly longer spans of time. Seven different surveys are used for this analysis, of which one captured on Oct. 10 is a reference to which the other six are aligned. Four fall surveys from Oct. 24, Oct 29, Nov. 7, and Nov. 14 represent approximately two-, three-, four-, and five-week time gaps. A summer survey from June 25 and a winter survey from Jan. 22 represent 3.5 month and 10 month time gaps. Rather than optimize for precisely aligned images that advance the comparison to the next 9.27m section of shore (see Section IV-D), we optimize for precisely aligned images closest to the two locations.

Figure 5 shows the results. We include the reference images from the Oct. 10 survey because images from different surveys best align to slightly different reference images. The number of each case of alignment quality is consistent with what's shown in Fig. 4. A precise alignment was found for A1-A3, A6, and B1-B3. Coarse alignments were found for A4, A5, and B4. B5 and B6 were the only two misalignments.

The natural variation in appearance between surveys mostly had little effect on the quality of the alignments. In A1, the colors are different, the weather is sunny instead of overcast, and the leaves are a different color or fallen. Although some artifacts are introduced near the shore of A5, both A5 and A6 align well. The same can be said for B1-B4.

The misalignments in B5 and B6 occurred because a large tree fell into the water between the surveys. During a comparison this should be seen as a "change." This change is, however, somewhat obfuscated because the aligned images capture different perspectives of the shore: in one case the boat swerves to miss the obstacle in the water, in the other the boat's path is unobstructed. Thus, for these two cases, a good alignment with the survey on Oct. 10 probably involves images of a slightly different location.

VI. Discussion

Whole images worth of descriptive features enable dense correspondence for natural environments. This contrasts the feature matching strategy of [12], which prefers a few highly descriptive features for sparse feature matching. Whereas too few sparse features may match between lakeshore surveys to enable data association along those lines [5], finding dense correspondence with the best feature candidates through whole image alignment provides robustness to natural variation in appearance. This is similar to the recent work of [13], which has found that matching patches of images, rather than point-based features, is better suited for outdoor environments. In contrast to their work, however, ours captures advantages of both cases, by optimizing for whole image correspondence derived from local feature matches.

That said, finding any correspondence between images is usually only possible if they have some common structure.

Salient structures anchor the correspondence between images. Everything else has a secondary contribution to the resulting alignment. A good example of images with salient structure is shown in Fig. 1. The opening along the shore is framed with strong tree-trunk features. Although locally individual features may appear somewhat different, globally a good correspondence between the images becomes visible. A good example of images without salient structure is shown in Fig. 5.B5. Perhaps the most salient features occur at the tree line, yet because the images are captured from different perspectives this is different between the images. Additionally, the reference image has strong features near the bushes (the tree in the water), which are hard to match in the aligned image.

This shows that changes between surveys can reduce the alignment quality. Small changes (e.g., people appearing) and changes with little structure in them (e.g., clouds) have less impact because salient structures of the scene still match. A precise alignment is less likely if a salient structure changes between two surveys (with other sources of variation) because as much information indicates an alignment is wrong as indicates is correct. However, because our system performs coarse-to-fine alignments, there is still a high likelihood of at least a coarse alignment between images.

Our coarse-to-fine approach makes image alignment possible. In this paper, visual SLAM provided the coarse alignments. If we tried to use SIFT Flow on its own, the computation time to run it (about 30 seconds per image) would be prohibitive. It is inefficient to compare every image between two surveys to find the one with the best alignment score. Additionally, although the alignment score is accurate most of the time, a low score is sometimes reported for images of different scenes. Coarse-to-fine alignment reduces this possibility. Therefore, the combined approach of visual SLAM and SIFT Flow makes robot–enabled lakeshore monitoring possible.

VII. Conclusion and Future Work

This paper introduced a framework for lakeshore monitoring, which enables a human to readily inspect images from two different surveys. The framework performs image alignment in a coarse-to-fine way, which bypasses many limitations arising from the large spatial scale. Dense, pixel-level correspondence is achieved between surveys, which eludes the temporal limitations inherent in data association across time in natural environments. An end-user who monitors a lakeshore using our framework can readily identify how a scene changed due to the flickering of images they compare. Overall, these analyses were validated using several surveys of a lakeshore captured over a long period of time.

The vast amount of data captured for this kind of study makes working with it, currently, the biggest limitation. In order to create analyses that span our entire dataset, in future work we plan to improve our framework in ways that reduce its computation time. For example, because each alignment is found independently of the others, the full computation time to perform SIFT Flow is required on all the images. This is inefficient because an alignment at one time step may be very similar to the following one. However, tying the search together across a sequence of images may reduce the total computation time, boost the alignment performance, and lead to more consistent flow image to image.

References

[1] Luca Carlone, Zsolt Kira, Chris Beall, Vadim Indelman, and Frank Dellaert. Eliminating Conditionally Independent Sets in Factor Graphs: A Unifying Perspective based on Smart Factors. In *ICRA*, pages 4290–4297, 2014.

[2] 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, 2013.

[3] Frank Dellaert. Factor Graphs and GTSAM: A Hands-on Introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sep. 2012.

[4] Arren J Glover, William P Maddern, Michael J Milford, and Gordon Fraser Wyeth. FAB-MAP+ RatSLAM: Appearance-based SLAM for multiple times of day. In *ICRA*, pages 3507–3512, 2010.

[5] Shane Griffith, Paul Drews, and Cédric Pradalier. Towards autonomous lakeshore monitoring. In *ISER*, 2014.

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

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

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

[9] Edward Johns and Guang-Zhong Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *ICRA*, pages 3212–3218, 2013.

[10] 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.

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

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

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

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

[15] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *ICRA*, pages 1678–1685, 2007.

[16] Harold Pashler. Familiarity and visual change detection. *Perception & psychophysics*, 44(4):369–378, 1988.