Towards Autonomous Lakeshore Monitoring

Shane Griffith, Paul Drews, and Cédric Pradalier

GeorgiaTech Lorraine - CNRS UMI 2958
Georgia Institute of Technology
{sgriffith7,pdrews3,cedric.pradalier}@gatech.edu

Abstract. This paper works towards autonomous lakeshore monitoring, which involves long–term operation over a large–scale, natural environment. Natural environments widely vary in appearance over time, which reduces the effectiveness of many appearance–based data association techniques. Rather than perform monitoring using appearance–based features, we are investigating whether the lakeshore geometry can provide a stable feature for this task. We have deployed an autonomous surface vessel 30 times over a duration of 8 months. This paper describes our initial analyses of this data, including our work towards a full simultaneous localization and mapping system and the shortcomings of using appearance–based features.

Keywords: lakeshore monitoring, SLAM, 3D reconstruction

1 Introduction

Efficiently monitoring a natural environment requires detecting and then exploring places that appear to be novel. With natural variation of appearance over short-, mid-, and long-term time scales, almost every location in an outdoor environment could be said to have changed, and thus, be a candidate for extensive exploration every time a robot is deployed there. The task of monitoring an expansive outdoor environment dictates a robot acquire an accurate model of the space. Yet, it is unclear how a robot could begin to acquire a model that would enable it to efficiently perform monitoring tasks.

Stationary monitoring tasks involve placing a camera in a predetermined location, which can simplify change detection because the camera is always pointing at the same spot and registering minor scene variation over consecutive frames. In this case, change detection is mostly unaffected by natural scene variation due to the high frame rate. Robotic monitoring, in contrast, involves a moving camera that captures intermittent snapshots of a scene. At first approximation, change detection following the stationary monitoring approach would require searching for images to compare and then finding an alignment between them. However, because the robot is capturing images of a non-planar surface, there usually is not a simple transform between images to align them. Furthermore, more variation accumulates between successive snapshots of a scene because the time interval between them is large.

Autonomous lakeshore monitoring calls for a representation that is stable across intermittent observations. The geometry of a lakeshore may be one such property. Some appearance–based features (e.g., SIFT, SURF) are invariant to many types of changes (e.g., scale, orientation, illumination), yet most are too unstable for comparing subsequent observations of a natural environment [7] [26]. However, it is the case that the appearance of many things in natural environments change with regularity, which may be possible to model in order to gain more predictive power. Thus, a robust representation of a lakeshore might use scene geometry as a basis for scene comparison, with spatiotemporal models of visual appearance supplementing it.

This paper starts to investigate the challenges of modeling a lakeshore environment using an autonomous surface vessel (ASV). Because a single sample of any outdoor environment is inadequate for capturing the distributions of its variations, we have deployed our robot for weekly data collection, so far 30 times. We are working towards applying SLAM techniques to extract an initial model of the lakeshore. This paper describes the challenges of using appearance–based features for data association between weekly surveys. This is ongoing work, in which we continue to collect data, improve our ASV system, and generate further analyses of our dataset.

2 Related Work

Autonomous lakeshore monitoring is a potential application of robotics for several reasons including the need to maintain water quality [11][19], monitor the environmental effects of dams [25], identify adverse uses of a lakeshore [25], and survey rare plants [21]. Beyond these applications this problem is also interesting for its theoretical and practical challenges. Acquiring a 3D model of a lakeshore presents significant difficulties to the predominant 3D reconstruction and mapping techniques.

Existing work on 3D reconstruction, structure from motion, and simultaneous localization and mapping (SLAM) provide a well–established framework for addressing how to map an unchanging environment using camera images captured by a robot. These techniques usually involve extracting features from each image, performing feature matching between images, triangulating the 3D position of features using the estimated camera poses, and then refining these estimates using non-linear optimization techniques. Agarwal *et al.* [1] used a framework based on this approach to reconstruct a 3D depiction of some monuments of Rome in a day using images freely available on the internet. Davison *et al.* [6] showed how a robot can map an environment using a sequence of images from only a single camera.

Some papers have addressed the specific challenges of learning in outdoor environments: over a large area, with multiple observations over an extended period of time, accumulating several experiences of the same location, and in a variety of lighting and weather conditions, which make them especially relevant to our work. Churchill and Newman [4] present a system that avoids data association and instead accumulates "multiple experiences" of scenes, which consist of images that are localized using visual odometry and landmarks. This style of representation may be worth exploring for using with lakeshore monitoring. Glover *et al.* [7] combine two techniques for performing SLAM using images of outdoor scenes captured at different time intervals. Their approach can map an outdoor environment in a way that is somewhat robust to scene variation, but it undesirably generates a lot of new descriptors for re-visited locations. The authors speculate that their method's shortcomings are due to the fact that mapping is grounded in highly variable appearance–based keypoints. Nourani-Vatani and Pradalier [20] use optical flow to reduce feature matching time. The optimal flow indicates the direction the robot is headed. This information is saved to a topology, which indicates what set of visual features in the database to use for comparison. Ki *et al.* [18] propose a divide-and-conquer strategy for scalable optimization. A map is divided into submaps, which are optimized independently, and then combined later into a global map. Procopio *et al.* [22] show that an autonomous robot can use near–field stereo data to learn a classifier for identifying far–field obstacles in images. An ensemble of classifiers makes the system more robust to wide variations in the visual appearance of different outdoor scenes.

Images of a lakeshore consist primarily of land, water, and sky, yet research has found that different features may be more suitable for each one. Specifically, the predominant approach for modeling things on land involves extracting visual features (e.g., SIFT [16]); yet, mounting evidence suggests different techniques may be better suited for capturing information about water [12][11][8]. Iqbal *et al.* [12] reveal that a primary difficulty of trying to establish a fundamental vision–based feature for water detection is due to all the possible sources of variation in a scene. Furthermore, without the context that land provides, there is a lack of visual features in open and deep water [11]. Instead, other sensory modalities can be a good source of information about water, including laser [13], and audio and proprioception [8]. Currently, our research is more focused on representing the visual appearance of the lakeshore, rather than that of the water or the sky.

The use of a collection of image processing techniques for scene analysis is supported by successes modeling sources of scene variation. Sources of variation include, for example, shadows and artifacts, and methods have been developed to individually address each one. For example, explicitly representing the source of illumination (and then removing its effects, e.g., [5]) may allow us to more easily analyze how the appearance of certain plants correlates with changing levels of sunlight. Additionally, modeling scene variation may be easier if we eliminate effects that aren't likely to be repeated across surveys, like image artifacts (e.g., due to water droplets or dust on the camera protector) [9].

Fig. 1. The Kingfisher as it traversed the perimeter of Lac Symphonie.

3 Experimental Setup

3.1 Robot

We used a Kingfisher from Clearpath Robotics for the experiments (see Fig. 1). The Kingfisher is an ASV propelled by a jet thruster in the end of each of its two pontoons. It is approximately 1.3 meters long and 0.9 meters wide, with space on top for optional sensors. Ours is equipped with an IMU, a compass, GPS, a forward–facing fish-eye camera, a top–mounted laser rangefinder, and a top–mounted pan-tilt camera. An onboard computer running ROS (see [23]) provides autonomous control, data logging, and communication for up to three hours on one charge.

3.2 Lake

We deployed the robot on Lac Symphonie in Metz, France, which is about 415 meters long and spans roughly 220 meters at its widest point (see Fig. 2). Its widest point is also the location of an island 131 meters long and 87 meters wide. Runoff water and a small tributary feed the lake while it drains into a nearby creek. Shrubs, bushes, trees, foliage, birds, and pedestrians abound. A fitness path and a scenic trail encircle the lake. Collegiate and technology buildings loom in the background.

3.3 Behavior

The robot moves along the perimeter of Lac Symphonie and then the island with its pan–tilt camera pointed at the shore. It traverses them in a counter-clockwise direction at an average speed of 0.5 m/sec. As it moves along the

Fig. 2. Lac Symphonie in Metz, France from Google Maps. The blue line is approximately the path the Kingfisher traversed around the lake, roughly 10m from the lakeshore.

shore, the laser rangefinder captures a scan of distances to it, which a motion planner uses to optimize the boat's behavior for maintaining its 10m distance. We chose 10m to keep the kingfisher distant enough from the shore to avoid tree branches and shallow water, yet close enough to capture fairly high resolution snapshots. A human intervenes using an RC controller if the boat gets too close to tree branches or fishing lines. The boat performs one survey in approximately an hour.

A survey along the perimeter of the lakeshore was performed as often as once per week over a duration of ten months. From August 18, 2013 to June 13, 2014, the robot traversed the perimeter of the lakeshore a total of 30 separate times. Data could not be collected during weeks the lake was frozen, in rainy weather, or if we were traveling.

3.4 Natural Scene Variation

The robot captured significant scene variation across the entire observation period, with many instances of large scene variation between consecutive surveys. Much of the variation in the dataset comes from natural variation in appearance over three seasons. The trees and the bushes changed color and shed leaves, revealing buildings and other landmarks behind them. Moles continuously burrowed new mounds out of the grass. Changing water levels turned grass to mud and destroyed some plants. The cloud cover varied, the water rippled in different ways, and shadows appeared in different places.

Another source of scene variation in the dataset is due to the fact that the robot takes a slightly different path around the lake each time. Fluctuating water levels raise and lower the robot, but also change its distance to things on the shore. Swans occasionally affected the boat's path if they floated past its starboard side. The robot also sometimes got as close as 2-3 meters away from shrubs near the shore that were too thin for the laser scan to consistently detect.

In the midst of natural scene variation, the most apparent changes in the scene were due to the activities of people. About half the surveys capture the construction of a new, shed–sized filtration building near the inlet to the lake. One survey captured canoes and kayaks on the shore for a water recreation event. Roughly a dozen fishermen held weekly competitions on Thursdays.

4 Methodology

4.1 Data Collection

This paper analyzes images primarily from two different surveys of the lakeshore. The boat's pose is estimated using GPS, IMU, and compass data. Images are captured as a sequence of 704 x 480 color images at 10 frames per second. A slight JPEG compression was applied to the images to increase the storage capacity of the boat.

4.2 Scene Reconstruction Using SIFT

We are working towards a monocular SLAM system for building a robust map of the environment, including dense scene reconstruction to capture the lakeshore geometry, data association across surveys, and optimization. This paper provides an analysis of scene reconstruction using SIFT.

Dense scene reconstruction is performed to precisely model the geometry of the lakeshore. The standard 3D estimation techniques provide a way to construct an estimate of the geometry from a sequence of images. Given matching points between two images, the 3D position is triangulated using Hartley and Thurm's iterative linear least–squares triangulation method [10]. Dense matches along a lakeshore enable the robot to more precisely estimate its geometry.

We initially used SIFT feature detection and matching [16] as provided by the OpenCV [2] library for acquiring a set of matches between pairs of images. Highly discriminative SIFT features (and other keypoint detection algorithms) are designed to be robust to large image transformations and changes in viewpoints. SIFT features are extracted from each image and then matches are found between pairs of images by comparing their feature sets. Because a lakeshore is, however, a mostly homogenous environment with very little change of viewpoint between frames, using SIFT can lead to a small number of matches, which are concentrated at high-contrast locations of each scene (e.g., buildings). Searching an entire feature set for each potential match can also make the process

Fig. 3. Snapshots of a scene along the lakeshore. The colored dots are the only features that could be matched. **top left** The result of feature matching for two nearby images in the same survey. **top right and bottom** The result of feature matching between this image and the top left image.

computationally inefficient. Our feature matching experiment illustrates these shortcomings.

5 Feature Matching Experiment

Our first experiment was useful for identifying the challenges of feature matching, both in within surveys and across different surveys. We applied our initial feature matching approach using SIFT to pairs of images of the same scene from the same survey, and to pairs images of the same scene from different surveys. Figure 3 shows the results. Many more matches are found between images from the same survey than from two different surveys. In the same survey condition, good matches are found on plants, the ground, and the building, but few are in the sky or in the water. Far fewer matches are found for images from two different surveys. The building is, however, the exception as its visual features are consistently matched across the surveys.

This result shows the difficulty of finding feature matches in images captured in natural environments at different times, even if they are of the same scene. With images of a lakeshore, the overall area of an image contributing to a successful match is small because SIFT features are only reliably detected on land. Additionally, the appearance of a natural scene in an outdoor environment

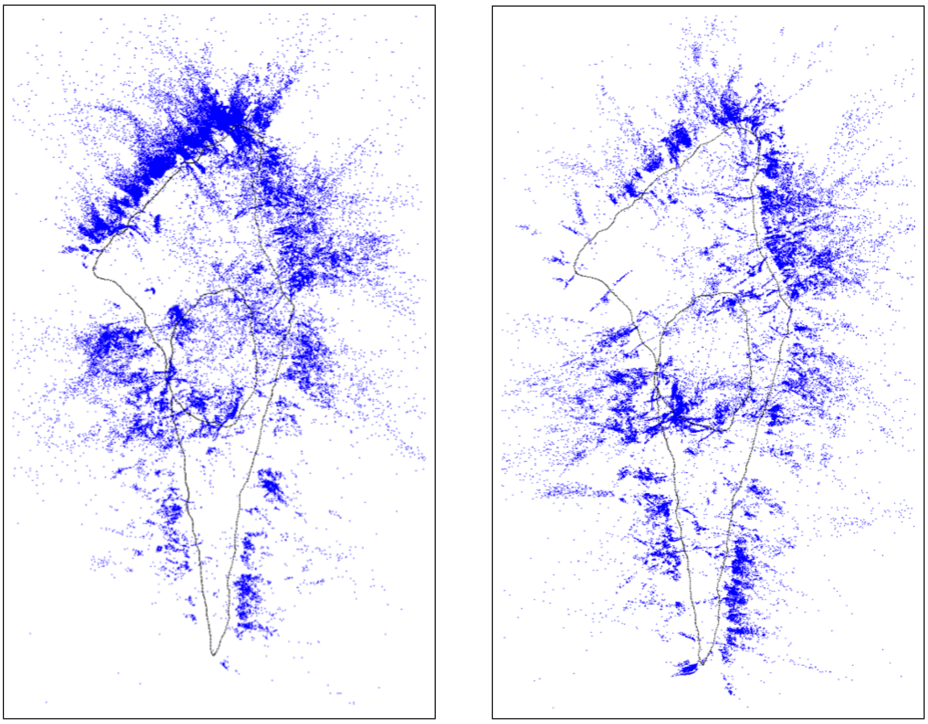

Fig. 4. Point clouds of **black** the boat's path around the lake, and **blue** the discriminative visual features in the scene, from two different surveys. Each image was generated using the Point Cloud Library [24].

can vary dramatically over any time scale, including abrupt changes in illumination, mid-term changes in water levels, and long-term seasonal variation in plants. Compared with flora, however, the contrast on buildings is more consistent across surveys, which is partly why many more features were found on the building.

Given that the number of feature matches on the lakeshore can be highly variable and that they are concentrated in high-contrast areas, we next identified the algorithm's coverage of the entire perimeter of the lakeshore. This analysis consisted of extracting features using SIFT, finding matches within the same survey, triangulating the 3D locations of matched features, and then visualizing the points in a point cloud, for an entire survey. The point clouds for two different surveys are shown in Fig. 4. Each has enough points to show that their structures are very similar. The density of each point cloud shows where feature matches are found best around the perimeter of the lake. Noise is apparent in a good portion of the 3D position estimates due to the fact that optimization has not yet been performed on these results.

The point clouds capture much of the 3D structure of the lakeshore, but due to the sporadic coverage of SIFT feature matching, some locations are better represented than others. Many points are identified on high-contrast areas like buildings, trees, and unique terrain. The point cloud is a bit thinner in one survey due to the overcast weather. In general, areas with fewer points are either not illuminated well, are part of the featureless grassy bank, or are not viewed by the camera. In the top-left of both surveys, the low-setting sun caused sun glare, which reduced the performance of SIFT. Due to these shortcomings of SIFT, we are currently implementing further improvements to our feature extraction method (namely, KLT, and optimization) for improved coverage of natural environments.

6 Ongoing Work

6.1 Scene Reconstruction Using KLT

In light of the shortcomings of SIFT, we have identified that the pyramidal Lucas–Kanade tracker [17] (LKT or KLT, OpenCV) provides an excellent feature–tracking performance for our environment and experimental setup. Instead of matching feature descriptors, the Lucas–Kanade tracker uses the brightness constraint and a smoothness assumption to compute sparse optical flow on features detected by the Harris corner detector.

In practice, building reliable feature tracks over a large image sequence requires a few more steps, some generic and others specific to our experimental setup. We try to sustain a stable number (200-300) of features in each image, while keeping them well spread over the landscape. Toward this, new feature candidates are extracted for each image and then sorted into the cells of a grid, as illustrated in Fig. 5. Features are only added to cells in which there are currently no features being tracked, and feature matches are only searched in a predicted neighborhood of cells. Thus, the number of features to track in each frame is limited, which ensures efficient computation time.

To limit the number of features ever tracked to a pre-specified maximum (300), a feature is removed from tracking when:

– OpenCV's Lucas–Kanade tracker cannot find a suitable match in the new image;
– its displacement between two frames is inconsistent with the robot's potential displacement (e.g., vertical displacement and large overall displacement);
– it moves into a cell in which multiple features (more than 5) have agglomerated (e.g., when features in the background are occluded);
– it is an outlier in a RANSAC–based fundamental matrix estimation.

The performance of feature tracking is depicted in Fig. 6. The number and the length of the black feature path shows that we can obtain reliable and stable features over fairly long sequences in a single survey. There are, however, several challenges associated with using this method for extracting dense coverage of a scene. A feature that is temporarily lost (e.g., if it is occluded or moves out of the field of view) and then reappears, for example, is not associated with

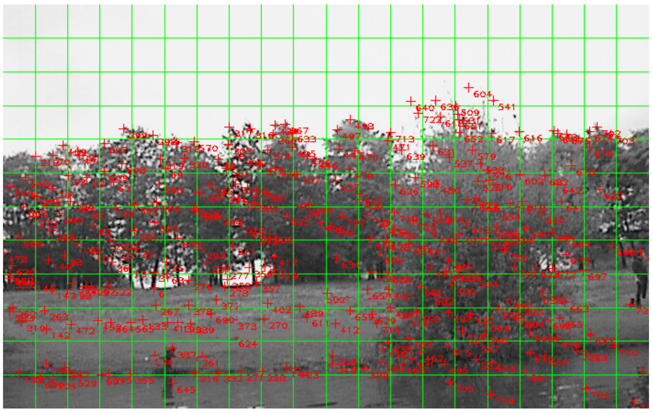

Fig. 5. Dense feature set and the grid used to enforce a relatively homogeneous feature distribution. The red numbers identify each feature and its ID.

its previous track in our implementation. Instead, a new track is created. Our current implementation does not address this limitation of KLT because it works well as-is.

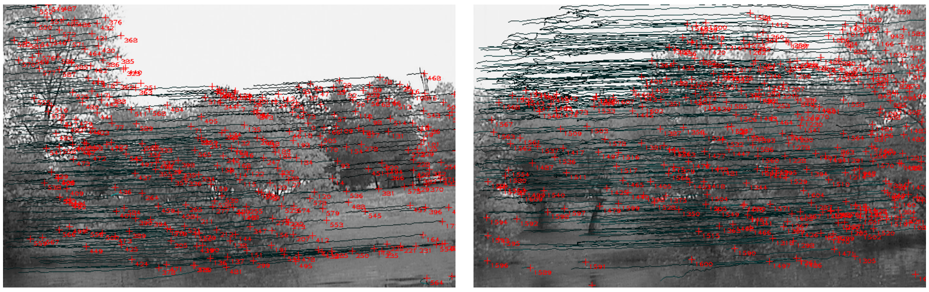

Fig. 6. Feature tracks (black) over two different sequences of 50 images. The red text identifies each feature and its ID. The length of each black line indicates the length of the feature track up that point.

6.2 Optimization

As the robot moves through the environment, it uses measurements of its motion and the projections of landmarks to compute estimates of the robot's poses and of the landmarks' positions, which are subsequently optimized using a non-linear optimization framework. Each measurement defines a constraint on the sequence of poses of the robot and the positions of landmarks. The values of each pose and landmark position are optimized using the correlated data from the other measurements. As the robot explores, it acquires an increasing number of different poses and landmark positions to estimate.

This paper applies the iSAM2 framework to perform optimization [14]. iSAM2 first captures the dependencies among the sequence of robot poses and the landmark positions using a factor graph. Pose and projection measurements are represented as factors in the graph, which define the constraints on robot poses and landmark positions to be optimized. The robot pose and the landmarks' 3D positions are represented as hidden variables to be estimated. In the iSAM2 framework, the factor graph is converted into a Bayes tree, first by applying variable elimination to convert the factor graph into chordal Bayes net, and then by extracting a directed tree from the cliques of the Bayes net.

Representing the optimization problem as a Bayes tree allows for incremental updating without having to solve the entire optimization problem. The cliques of the Bayes tree affected by a new factor are removed and deconstructed into a factor graph. The variables of the new factor graph are reordered and converted back into a bayes tree, which is placed at the root of the unaffected portion of the deconstructed tree. For nonlinear factors, an additional relinearization step is performed to keep a valid linearization point for each variable.

To supplement iSAM2, we also utilize *smart factors* to reduce the computation time required for optimization [3]. Because we are applying SLAM in a long–term monitoring application, a very large number of factors accumulate to represent all the robot poses and the projection measurements of landmarks, and each directly increases the computation time required for optimization. It is possible, however, to reduce the number of factors used for optimization by taking advantage of conditional independence relations of landmark observations. Each landmark observation provides 'support' data, which is used for helping estimate a robot pose and a landmark's location. Fortunately, a set of support variables for one landmark is conditionally independent of a set of support variables for a different landmark given a *smart factor*.

A smart factor provides an abstraction of the data observed for a landmark. This is enabled by the Schur complement, which reduces a system of linear equations from one large problem into several smaller subproblems. As long as each subproblem is well–conditioned, solving these subproblems is equivalent to solving the large one. Because a smart factor abstracts all the projection data for a landmark, it can also seamlessly check that its subset of the data is well-conditioned. Degenerate cases like rotation–only movement, movement towards a landmark, and a single observation of a landmark are eliminated before they are used for optimization.

7 Conclusion and Future Work

Natural environments are challenging for the predominant SLAM data association techniques. Towards our long term goal, this initial paper found that SIFT, which is an appearance–based feature, provides spotty coverage of natural scenes and does not match well across surveys. These results are consistent with similar work on mapping in large natural environments over large time scales. In light of this, we have identified that KLT in addition to iSAM2 optimization may provide a more precise estimate of the geometry of a lakeshore.

This is ongoing work in which we are continually gathering data, improving our approach, and generating results. As we continue to get results, we are looking toward data fusion (using geometry for data association) in the optimization framework to help create a more comprehensive map of the lakeshore, as in [15]. We also plan to reduce the noise in robot pose estimates by applying the iterative closest point algorithm to the laser scan of the shore for improved odometry. Because this is still very preliminary work, we expect our methodology, analyses, and conclusions to substantially grow over time.

Acknowledgements

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

References

1. Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S.M., Szeliski, R.: Building rome in a day. Communications of the ACM 54(10), 105–112 (2011)
2. Bradski, G., Kaehler, A.: Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc. (2008)
3. Carlone, L., Kira, Z., Beall, C., Indelman, V., Dellaert, F.: Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on (2014)
4. Churchill, W., Newman, P.: Experience-based navigation for long-term localisation. The International Journal of Robotics Research 32(14), 1645–1661 (2013)
5. Corke, P., Paul, R., Churchill, W., Newman, P.: Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. pp. 2085–2092. IEEE (2013)
6. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29(6), 1052–1067 (2007)
7. Glover, A.J., Maddern, W.P., Milford, M.J., Wyeth, G.F.: Fab-map+ ratslam: appearance-based slam for multiple times of day. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on. pp. 3507–3512. IEEE (2010)
8. Griffith, S., Sukhoy, V., Wegter, T., Stoytchev, A.: Object categorization in the sink: Learning behavior–grounded object categories with water. In: Proc. of the 2012 ICRA Workshop on Semantic Perception, Mapping and Exploration (2012)
9. Gu, J., Ramamoorthi, R., Belhumeur, P., Nayar, S.: Removing image artifacts due to dirty camera lenses and thin occluders. ACM Transactions on Graphics (TOG) 28(5), 144 (2009)
10. Hartley, R.I., Sturm, P.: Triangulation. Computer vision and image understanding 68(2), 146–157 (1997)

11. Hitz, G., Pomerleau, F., Garneau, M.E., Pradalier, C., Posch, T., Pernthaler, J., Siegwart, R.Y.: Autonomous inland water monitoring: Design and application of a surface vessel. Robotics & Automation Magazine, IEEE 19(1), 62–72 (2012)
12. Iqbal, M., Morel, O., Meriaudeau, F., Komputer, F.I.: A survey on outdoor water hazard detection. In: The 5th Intl. Conf. on Information and Communication Technology and Systems (ICTS) (2009)
13. Jain, S., Nuske, S.T., Chambers, A.D., Yoder, L., Cover, H., Chamberlain, L.J., Scherer, S., Singh, S.: Autonomous river exploration. In: Field and Service Robotics, Brisbane (December 2013)
14. Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J.J., Dellaert, F.: isam2: Incremental smoothing and mapping using the bayes tree. The International Journal of Robotics Research 31(2), 216–235 (2012)
15. Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy, N., Teller, S.: Multiple relative pose graphs for robust cooperative mapping. In: Robotics and Automation (ICRA), 2010 IEEE International Conference on. pp. 3185–3192 (2010)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
17. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. In: IJCAI. vol. 81, pp. 674–679 (1981)
18. Ni, K., Steedly, D., Dellaert, F.: Tectonic sam: Exact, out-of-core, submap-based slam. In: Robotics and Automation, 2007 IEEE International Conference on. pp. 1678–1685. IEEE (2007)
19. Niinioja, R., Holopainen, A.L., Lepistö, L., Rämö, A., Turkka, J.: Public participation in monitoring programmes as a tool for lakeshore monitoring: the example of lake pyhäjärvi, karelia, eastern finland. Limnologica-Ecology and Management of Inland Waters 34(1), 154–159 (2004)
20. Nourani-Vatani, N., Pradalier, C.: Scene change detection for vision-based topological mapping and localization. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. pp. 3792–3797. IEEE (2010)
21. Pavlovic, N.B., Bowles, M.L.: Rare plant monitoring at indiana dunes national lakeshore. Science and ecosystem management in the national parks. University of Arizona Press, Tucson pp. 253–280 (1996)
22. Procopio, M.J., Mulligan, J., Grudic, G.: Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments. Journal of Field Robotics 26(2), 145–175 (2009)
23. Quigley, M., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source Robot Operating System. In: ICRA workshop on open source software (2009)
24. Rusu, R.B., Cousins, S.: 3d is here: Point cloud library (pcl). In: Robotics and Automation (ICRA), 2011 IEEE International Conference on. pp. 1–4. IEEE (2011)
25. Single, M.: Mokihinui hydro proposal consent applications review of assessment of effects at the coast. Technical Report (2008)
26. Valgren, C., Lilienthal, A.J.: SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. Robotics and Autonomous Systems 58(2), 149–156 (2010)