

Perception for a River Mapping Robot

Andrew Chambers, Supreeth Achar, Stephen Nuske, Jörn Rehder, Bernd Kitt,
Lyle Chamberlain, Justin Haines, Sebastian Scherer, and Sanjiv Singh

Abstract—Rivers with heavy vegetation are hard to map from the air. Here we consider the task of mapping their course and the vegetation along the shores with the specific intent of determining river width and canopy height. A complication in such riverine environments is that only intermittent GPS may be available depending on the thickness of the surrounding canopy. We present a multimodal perception system to be used for the active exploration and mapping of a river from a small rotorcraft flying a few meters above the water. We describe three key components that use computer vision, laser scanning, and inertial sensing to follow the river without the use of a prior map, estimate motion of the rotorcraft, ensure collision-free operation, and create a three dimensional representation of the riverine environment. While the ability to fly simplifies the navigation problem, it also introduces an additional set of constraints in terms of size, weight and power. Hence, our solutions are cognizant of the need to perform multi-kilometer missions with a small payload. We present experimental results along a 2km loop of river using a surrogate system.

I. INTRODUCTION

We are developing a minimal sensor suite to be used by a low-flying aircraft to autonomously explore rivers, mapping their width and the surrounding canopy. In some cases, the canopy can be so thick and high covering a river that it blocks GPS signals and the problem becomes one of simultaneous localization and mapping in an unstructured fully three-dimensional environment. Exploration from a low-flying vehicle is attractive because it extends the sensing horizon and removes complications of navigating in shallow water and aquatic vegetation. However, a flying solution also adds constraints on the size, weight and power available for perception. This is a significant constraint given that the multi-kilometer missions will force all the sensing/computation to be conducted onboard. It is our estimate that given the size of rotorcraft that could reasonably fly in environments with thick canopy, it will be necessary to keep all the sensing and computation components to less than one kilogram.

These constraints on payload and the inability to rely on GPS have significant implications for our design. First, we will need to depend on perception to produce a high resolution 6DOF pose estimate that is much more stable than can be produced by simply integrating inertial sensors.

A. Chambers, S. Achar, S. Nuske, J. Rehder, L. Chamberlain, J. Haines, S. Scherer, and S. Singh are with The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., {achambers, ssingh}@cmu.edu

B. Kitt is with The Institute of Measurement and Control Systems, Karlsruhe Institute of Technology, Karlsruhe, Germany, bernd.kitt@kit.edu

The work described in this paper is funded by the Office of Naval Research under grant number N00014-10-1-0715.



Fig. 1: A typical riverine environment that we expect to map. A small rotorcraft is able to fly above the shallow, fast-moving water and yet remain below the thick canopy to navigate and map the river. The foliage along the banks is dense enough to block or seriously degrade GPS signals.

Second, any active imaging, such as from laser scanning, will be required to be very lightweight and low power and hence will be short range. Third, river-following without a prior map will require a perception system that goes significantly further than could be sensed through laser ranging.

In this paper we describe three key modules for perception. The first is a short range, laser-ranging based system tasked with obstacle detection and the creation of a metric, three-dimensional map. Here we use a continuously rotating, lightweight line laser scanner to produce a three-dimensional scanning pattern that enables collision avoidance and the creation of a three-dimensional representation of the riverine environment. The second is a long range color vision system that uses a forward pointing color camera to automatically find the river even in the face of significant variation in the appearance of the river. We solve this problem with a self-supervised method that continually learns to segment images based only on an estimate of the horizon (from inertial sensing) and some simple heuristics that describe riverine environments. The third is a graph-based system for state estimation of the vehicle's motion in 6DOF. Using consecutive and global constraints from visual odometry, inertial sensing, and sparse GPS, we demonstrate its ability to globally estimate the vehicle's state while maintaining an accuracy that allows for precise local mapping.

The remainder of this paper is organized as follows. Section II reviews related work. Section III details the short-

range perception strategy. Section IV presents our method for visually identifying the river. Section V describes our solution for state estimation using visual odometry, IMU, and sparse GPS. Section VI discusses the approach to registering sensor data into global river maps. Section VII presents simulated and experimental results. Finally, section VIII concludes and discusses future work.

II. RELATED WORK

Previous work in autonomous river mapping has utilized small boats [1] or higher altitude, fixed wing UAVs [2]. While these platforms could be more practical in simple riverine environments, we aim to develop a platform that can perform in the most difficult situations such as rapidly flowing water, obstructed waterways, or dense forest canopies. Preliminary work in river mapping on small rotorcraft using passive vision and ultrasonic ranging (for elevation estimation) has been reported over short distances [3]. Our work is similarly motivated but we explicitly consider substantially longer missions in which it is important to not only map the extent of the river but also to map the vegetation along the shore line and avoid obstacles that might appear in the middle of the river.

The three main sections of our work are river detection and mapping using passive vision, position estimation for a lightweight flying vehicle in areas of sparse GPS availability, and 3D mapping.

In the first area, our system uses a visual river segmentation algorithm to detect the extent of the river for mapping and guidance. Most previous work on detecting water in images has been focused on detecting water hazards like puddles using color, texture and stereo disparity cues for autonomous ground vehicles [4], [5]. Our problem is complicated by ripples on the water surface and the lack of stereo imagery captured from a sufficiently wide baseline. Our solution automatically learns models of river and shore appearance for segmentation by exploiting the structure of riverine environments in a scheme that shares some similarities to self-supervised road detection [6].

In the second area of position estimation, there has been a growing interest recently in SLAM for unstructured outdoor environments [7], [8]. Popular approaches employ an EKF to combine heterogeneous sensor inputs to estimate the vehicle state [9], [10]. In order to avoid issues related to linearization in EKF-based state estimation [11], we take an approach most similar to [12], [13], [14] that treats the problem as a nonlinear optimization represented by a graph. The approach successively linearizes around the current state, and is therefore able to deal with highly nonlinear sensor models.

In the third area of 3D mapping, our work is related to [15] where points are globally registered from a spinning laser scanner while the vehicle is moving. However, on an aerial vehicle we cannot use wheel encoders and instead use visual odometry as a primary input to the state estimation.

III. SHORT RANGE PERCEPTION

The rotorcraft must be able to operate in the space between the water's surface and the tree canopy. In this

cluttered space, a reliable short range perception system is necessary for obstacle avoidance and mapping. To measure 3D information about the environment, we use an off-axis rotating 2D laser line scanner. As seen in Fig. 2, the Hokuyo UTM-30LX is mounted with the scan plane tilted at 45° with respect to a sweep motor axis.

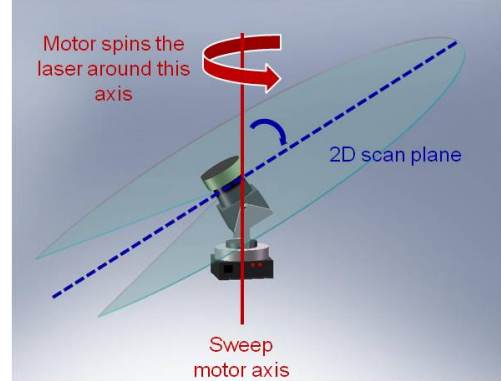


Fig. 2: Off-axis spinning scanner

Other laser mounting and actuation configurations such as nodding, spinning on-axis, or roundly swinging [15], did not provide the same scan density or sensing field of view. Our configuration has the advantage of equal detection of horizontal and vertical obstacles with a full 360° field of view. Fig. 3 shows the hatched scan pattern sensed by the spinning scanner. This scan pattern detects thin horizontal and vertical obstacles equally well as opposed to a nodding laser, which has difficulty detecting thin horizontal obstacles or an vertically-mounted on-axis spinning laser, which has difficulty detecting thin vertical obstacles. In a natural river environment, thin horizontal and vertical tree branches should be expected and can be reliably sensed with our configuration.

A. Registration and Calibration

To register ranged data points P_r from the laser into a non-spinning base frame P_b , we apply a series of 4x4 homogeneous transformations to each ranged point.

$$P_b = T_w T_m T_s P_r \quad (1)$$

T_w : Time dependent single axis rotation to account for the rotation of the laser module around the sweep motor axis. This angle is calculated by assuming a constant sweep speed and interpolating the angle given by the motor encoder at the beginning and end of the scan.

T_m : Full 6 DOF transformation, which remains constant and represents the mounting configuration of the scanner. This transform takes into account the tilt of the laser scan plane, the translation between sweep axis of rotation and the laser receiver, and any mechanical misalignment. We currently hand measure this transformation but are working on a nonlinear optimization similar to [16] or [17].

T_s : Time dependent single axis rotation, which expresses the position of the rotating mirror within the laser scanner.

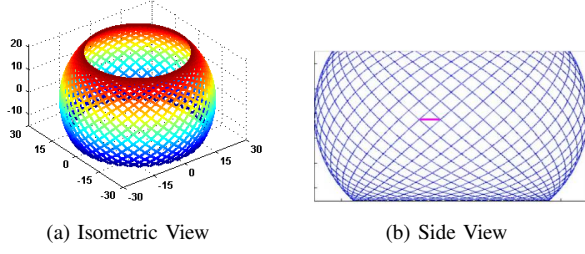


Fig. 3: Laser scan pattern when the scan plane is tilted at 45° to the sweep motor axis. The horizontal magenta line segment in 3b shows the widest unseen visual angle at the hatched diagonals.

B. Obstacle Detection Confidence

We developed a model of the laser's obstacle detection performance to provide a confidence measure for the detection of variously sized obstacles at different vehicle velocities. Obstacles are modeled as thin, floating objects to model the worst case scenario such as a hanging vine or thin branch. The magenta line in Fig. 3b could represent a horizontal obstacle. Obstacles are defined only by their length, which is a valid model for the small tree branches that will pose significant danger to the rotorcraft above the river. We make two important assumptions: 1) if a scan line falls on the object, the object will be detected and 2) the rotation of the hatched scan pattern varies randomly from sweep to sweep. This random rotation variation could come from external perturbations in the vehicle's yaw or could be manually introduced by adding an angular offset to the motor for each sweep. The unseen angle between scans after one sweep is:

$$\theta_u = \sqrt{2} \frac{2\pi\lambda_w}{\lambda_s} \quad (2)$$

where λ_w is the sweep rate and λ_s is the scan rate. The factor of $\sqrt{2}$ expresses the greatest unseen angle, which is at the diagonal of the square in the hatched scan pattern. The visual angle θ_o of an obstacle with length l at a distance r is expressed as:

$$\theta_o = 2 \arctan\left(\frac{l}{2r}\right) \quad (3)$$

The probability of detection after one sweep is simply the ratio of the obstacle's visual angle to the unseen angle, with a maximum probability of one. Here we consider a static obstacle directly in the path of the vehicle moving at velocity v from an initial starting distance D_0 . The distance r and thus the visual angle will be reduced for each new sweep. The probability of an observation after k sweeps is:

$$P_k(obs) = 1 - \left(1 - \min\left(\frac{\theta_o(r)}{\theta_u}, 1\right)\right)^k \quad (4)$$

$$r = D_0 - \frac{(i-1)v}{\lambda_w} \quad (5)$$

where the object's visual angle θ_o depends on r .

For a desired observation confidence, safe maximum velocities are found which satisfy the following equation with a fixed C-Space expansion L_C , reaction time t_a , and maximum vehicle acceleration a :

$$D_0 - L_C \geq \left(\frac{k}{\lambda_w} + t_a\right)v + \frac{v^2}{a} \quad (6)$$

The inequality states that the rotorcraft must observe the obstacle, react, and come to a stop in a distance equal to or less than the maximum laser range minus the C-Space expansion. Safe velocities are solved numerically using realistic obstacle sizes and presented in section VII.

IV. LONG RANGE PERCEPTION

The rotorcraft needs to be able to determine the course of the river and follow it. Also, river width measurements are needed to build the river map. The onboard laser sensor has an effective range limit of 30 meters which is sufficient for obstacle avoidance but not for guidance or mapping. Our solution is to use images from an onboard camera. Images are segmented to find the extent of the river from the current viewpoint of the vehicle. Using knowledge of the vehicle's orientation and height above the river provided by the vehicle's IMU and altimeter, the extent of the river in the image can be projected into a local coordinate frame. This forms a local map of the river for guiding the vehicle along the river's course. Many such local maps can be fused as the vehicle moves to build a global map of the river shape.

The main challenge is building a suitable appearance model for the water. Within a single image, the appearance of water can vary greatly due to reflections of the foliage and other structures on the bank, reflections of the sky, and dark regions that fall in the shadows. In addition, ripples on the water's surface create variations in texture. This variability in river appearance in an image makes it hard to determine the extent of the river. Also there are large changes in water appearance with variation in weather and illumination conditions. All these factors make it difficult to learn a single classifier to detect river regions that performs well in a variety of settings and which can generalize to previously unseen environments.

We build upon our earlier river segmentation algorithm [18]. The algorithm utilizes knowledge about the position of the horizon in each image and assumes that anything appearing above the horizon line is not part of the river. The image is divided into small square patches and a feature descriptor with color and texture information is calculated for each patch. Two generative appearance models are learned online from the input image, one for patches above the horizon (which are all part of the non-river region) and another for patches below the horizon (the majority of which we expect to be part of the river). These two appearance models are used to calculate a probability of each patch being part of the river. Patches with a high probability of belonging to the river and patches above the horizon line are used to train a two class (river/non-river) linear SVM which is then used to find the extent of the

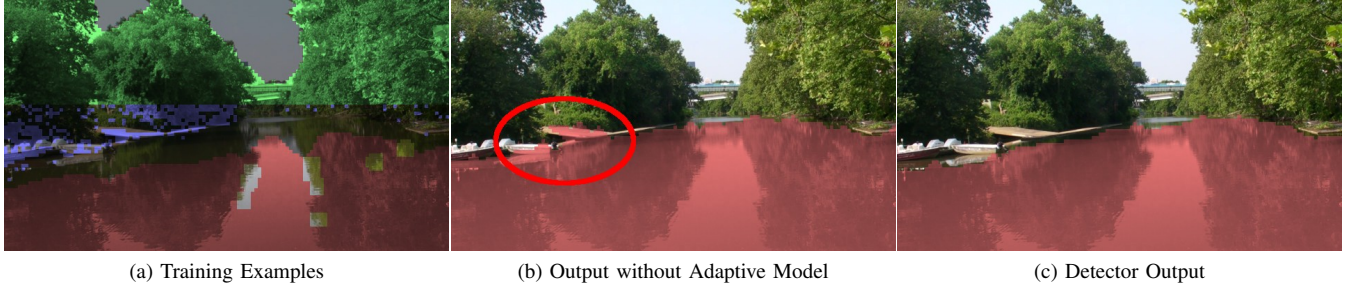


Fig. 4: Visual River Detection: (a) Input image with automatically selected training examples marked. The green regions above the horizon line are used as non-river example patches. The red regions are automatically selected as candidate river patches by the appearance modeler. The blue regions below the horizon are novel regions that did not fit well with the historical model of river appearance. The other highlighted patches below the horizon are places where reflections were detected. (b) The output of the river detection algorithm without the adaptive appearance model or reflection modelling. This version of the algorithm was completely memoryless and so it misclassified the pier and boats along the shore (circled). (c) The output of the river detector with the extent of the river marked in red. With the adaptive modelling, the algorithm is able to detect novel objects and classifies the pier and boats correctly.

river in the entire image. The original algorithm did not explicitly model reflections and was completely memoryless which made it difficult to correctly classify novel objects appearing below the horizon. We addressed these issues by detecting reflections and using a history of river appearance from previous frames to build an adaptive appearance model. These improvements are described in detail below.

A. Detecting Reflections

Reflections are a useful cue for determining the extent of the river. If an object and its reflection can be found it is fairly safe to assume that the reflection was from the water’s surface. We look for reflections of salient points (Harris Corners [19]) that occur above the horizon line. The water surface is modelled as a ground plane and the vehicle pose estimate from the state filter provides the orientation and distance to the plane. With this, it is easy to use the geometry of reflections to find where the reflection of a point in 3D space should appear. Since we do not have complete depth information, we instead localize each salient point’s reflection to a small region below the horizon where (if present) it should be found. We neglect effects caused by change in effective viewpoint between the object and its reflection and search this region for patches that are similar to a vertically mirrored version of the patch centered around the point being considered. The similarity measure used is normalized cross correlation. Pairs of patches that get similarity scores above a threshold are marked as object-reflection pairs. Patches surrounding the reflection are added to the SVM training data as examples of river patches.

B. Adaptive Appearance Modelling

In [18], new appearance models for the river and non-river regions were built from scratch for every new frame, the algorithm was unable to handle novel objects like piers or boats that appeared below the horizon because of the assumption that the above horizon part of the image is representative of the entire non-river region. To solve this problem we maintain two appearance models for the river, one built

as before from just the current frame, P_{cur} and another built from patches sampled from the river region segmented out in previously images P_{old} . Patches are sampled selectively from previous frames to build P_{old} , with patches from older images being less likely to be used than patches from more recently seen images. The probability of a patch from the k^{th} frame before the current frame being sampled for P_{old} is λ^k where $\lambda < 1$ is a parameter that sets the rate of decay. We define a novelty value $\eta(X)$ for each patch whose value is given by

$$\eta(X) = \log \frac{P_{cur}(X)}{P_{old}(X)} \quad (7)$$

$\eta(X)$ is a measure of how well a patch fits into the model of water appearance built over recent frames. A high value of $\eta(X)$ for a patch below the horizon in the current image means the patch is unlikely to be part of the river given the water’s appearance over the last few frames. We threshold $\eta(X)$ to select novel patches below the horizon line that are then used along with the above horizon patches as negative training examples in the SVM classifier. Figure 4a shows some of the novel patches below the horizon that the algorithm found.

V. VEHICLE STATE ESTIMATION

An estimate of the vehicle’s pose is required for flight control of the rotorcraft as well as mapping of the environment. In many applications, this estimate heavily depends on GPS measurements. However, in river mapping applications, GPS may not be available or unreliable for large stretches due to occlusion by the canopy. In order to deal with intermittent GPS signals, additional means of localization have to be exploited. Our approach relies on a filter architecture that combines visual odometry, inertial measurements, and sparse GPS to provide a combined state estimate that minimizes drift.

We use the approach to visual odometry presented in [20]. This approach makes use of the stereo setup by triangulating 3D points based on matches in one stereo pair. The relative

motion is determined by iteratively adjusting the translation and rotation of the camera in order to minimize the reprojection error of these 3D points in the consecutive image pair. A framing Kalman Filter with a constant acceleration model smooths the estimated motion and provides an error covariance for each pose.

In order to fuse inertial measurements, visual odometry and intermittent GPS readings, we employ a graph-based optimization approach comparable to [12], [21], [13] that estimates a set of vehicle states $[s_1, \dots, s_n]$, for which a set of measurements $[x_1, \dots, x_m]$ obtained with the onboard sensors was most likely to occur. In this framework, nodes represent the state s_k of the vehicle at different points in time t_k . Sensor readings x_i induce constraints on these states, which are represented as edges that connect two state nodes in the graph. We make the assumption that the measurement x_i can be modeled as $x_i = h_i(\hat{s}_{k_i}, \hat{s}_{l_i}) + v_i$, where $h_i(\hat{s}_{k_i}, \hat{s}_{l_i})$ is generally nonlinear function of two true states $\hat{s}_{k_i}$ and $\hat{s}_{l_i}$ with $k_i, l_i \in [1, \dots, n]$ and v_i is a zero-mean Gaussian distributed random vector with covariance matrix C_i . With $C_i^{-1} = Q_i^T Q_i$, the maximum likelihood estimation of the set of vehicle states is the one, which minimizes the following nonlinear cost function:

$$[s_1, \dots, s_n] = \arg \min_s \sum_{i=1}^m \|Q_i (x_i - h_i(s_{k_i}, s_{l_i}))\| \quad (8)$$

Visual odometry and integrated gyroscope measurements provide relative pose measurements, which result in constraints on consecutive nodes in the graph. GPS and the accelerometer as inclinometer impose global constraints on the state, which are represented by edges to a fictitious zero state at the origin of the global coordinate frame. The overall state S of the system is determined by collecting the state of each node and the cost E associated with this state is a vector containing the terms of all edges. We minimize the cost iteratively using the Levenberg-Marquardt algorithm [22]. With the Jacobian $J_E = \delta E / \delta S$, the state update ΔS is found by solving the augmented normal equation:

$$(J_E^T J_E + \lambda I) \Delta S = -J_E^T S \quad (9)$$

The sparse interconnection of the state nodes in the graph directly corresponds to a sparsity pattern in J_E , that we exploit by using an implementation of the LM algorithm optimized for sparse systems [23]. Furthermore, the optimization is only performed over a sliding window since the effect of constraints between distant nodes generally decays quickly along the graph.

VI. MAPPING

The previous sections provide an overview of how our robot will perceive and navigate the river. The next stage is to take the sensor data and produce metric maps of riverine environments. Our design will produce two types of maps, a 2D map of the river's shape and a detailed 3D point cloud of vegetation and canopy clearance.

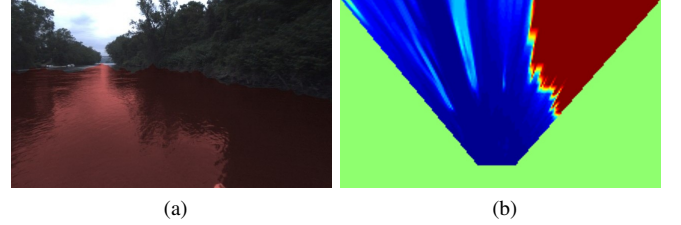


Fig. 5: Local 2D River Mapping: (a) Image with the detected extent of the river (thresholded at $P_{river} = 0.5$) overlaid in red. (b) The same detected extent of the river projected into a top down view. Blue regions are part of the river and red regions are the shore. The surrounding light green area is unobserved in this image as it lies outside the viewing frustum.

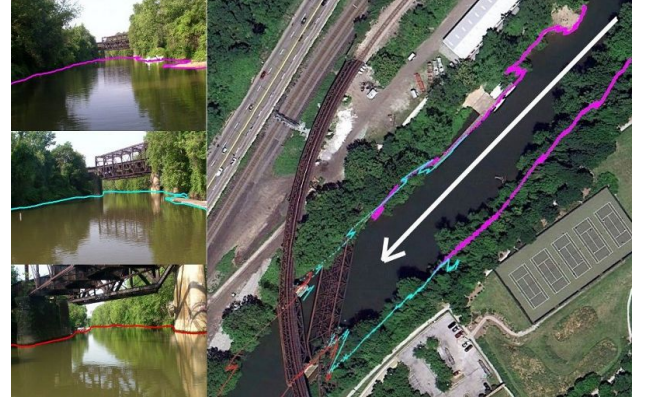


Fig. 6: Demonstrative example of 2D river maps our system can generate. Left: The three input images with the contour of the river highlighted. Right: The contours are projected from image into world frame, then imported and displayed on a satellite image.

A. 2D Map

Our system builds a 2D map of the river by registering the outputs of the visual river detection algorithm into a global frame using vehicle pose estimates from the state filter. The visual river detection algorithm outputs an estimate of the probability of each pixel in an image being part of the river or the shore. Since the river surface is assumed to lie on the ground plane, the probabilistic labeling from the vehicle's viewpoint can be projected into a top down view using knowledge of the vehicle orientation and height provided by the state filter. Each of these top down views is a local, vehicle centered map of a small section of the river (Fig 5b). These local maps are registered globally with position information from the state filter and fused into a global map of the river using occupancy grid mapping. Figure 6 shows a demonstrative example of the sort of map our mapping technique can generate. In satellite imagery, parts of the river are hidden under the canopy, but by mapping with a low flying camera we can see all of the river. Figure 11a shows a larger map built by our algorithm.

B. 3D Map

To generate a 3D map of the river environment we register laser scans from the spinning scanner into the world frame

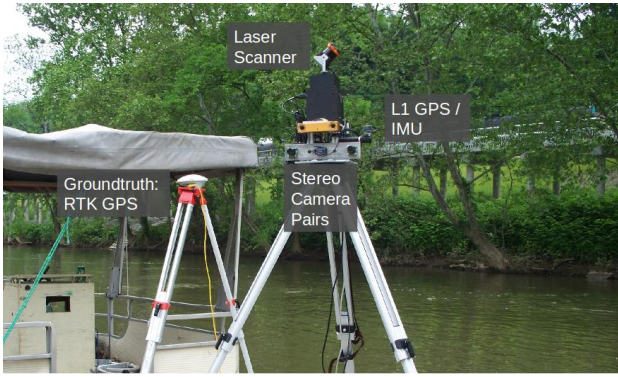


Fig. 7: Data Collection System

using the previous discussed state estimation framework. The approach requires us to calibrate the stereo camera pair with the spinning laser scanner and synchronize the data streams coming from both. The calibration between laser and camera is determined using the Laser-Camera Calibration Toolbox [24]. To time synchronize camera images, laser range data, and laser angular position, a microcontroller measures a rotary encoder attached to the spinning laser scanner and calculates the temporal offset between laser scans (marked by a 40 Hz pulse from the Hokuyo) and a 15 Hz camera shutter synchronization pulse.

VII. RESULTS

A. Short Range Perception

Fig. 8 plots safe rotorcraft velocities versus obstacle length from 0.1 m to 1 m for three different maximum laser ranges. Obstacle detection confidence is 95%, C-Space expansion is 0.5 m, reaction time is 0.1 s, maximum acceleration is $5m/s^2$, scan rate is 40 Hz and sweep rate is 1 Hz. Maximum velocities increase in discrete steps since we require that a full sweep must be completed before an obstacle can be detected.

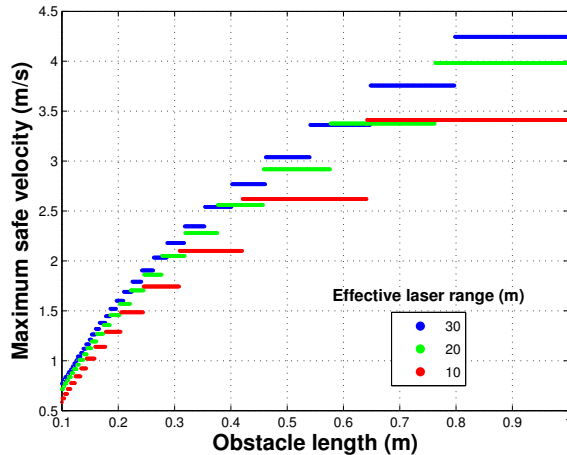


Fig. 8: Worst case analysis of maximum safe velocity for varying obstacle lengths. Maximum velocities increase in discrete steps since we require that a full sweep must be completed before an obstacle can be detected.

B. Long Range Perception

We tested the river detection algorithm using four datasets captured on two different rivers. Each dataset contains between 120 to 150 images taken from a boat moving along the river along with ground truth segmentation of the river region. These datasets were used in [18] to evaluate the earlier memoryless version of the algorithm. The performance of both versions of the algorithm are reported in Table I (error rates for the earlier memoryless and current adaptive versions are under the ‘old’ and ‘new’ columns respectively). The performance metric used for evaluation is the percentage of pixels misclassified by the algorithm compared to ground truth. The algorithm performs better with the reflection detector and the adaptive model of river appearance.

TABLE I: River Detector Performance: Mean Error Rates

Dataset	Old	New
Allegheny Day	2.75%	2.68%
Allegheny Dusk	4.15%	3.03%
Allegheny Fall	2.73%	2.05%
Youghiogheny	3.22%	3.15%

C. Vehicle State Estimation

Fig. 9 depicts the reconstructed path for various sensors overlaid onto an aerial map of the area. For ground truth, we acquired position information with a high accuracy L1/L2 GPS post-processed with RTKLIB [25], which is shown in green. Ground truth is not available when passing under the steel bridges since the GPS signal is lost. The sequence spans about 1.9 km and roughly 10,000 video frames.

Fig. 9a shows the results of the visual odometry for stereo image pairs taken at 15 Hz. The stereo camera is a Point Grey Bumblebee2 with an image resolution of 1024 x 768, 12 cm baseline, and 97° HFOV. For this sequence, the path sensed by visual odometry is smooth and locally accurate. However as errors accumulate, it diverges from the reference path quite significantly. The path in Fig. 9b was estimated by fusing visual odometry and inertial measurements recorded at 100 Hz. The IMU is a MicroStrain 3DM-GX3-35 with integrated L1 GPS. Gyroscope outputs constrain relative rotations, while the accelerometer served as an inclinometer to determine the orientation in a global coordinate frame. Incorporating additional sensor information improves the results, but errors still accumulate as the sensors provide mostly relative motion information. In Fig. 9c, position information from the cheap, lightweight, integrated L1 GPS receiver was incorporated into the estimation at a rate of 0.2 Hz. For most parts of the sequence, the resulting estimation lines up well with the reference path. Furthermore, the path is locally smooth, thus providing a suitable basis for laser point cloud registration. The motion estimated by visual odometry suffers from a scaling issue, which is also observed by [26]. We believe the underestimation of motion comes from a small stereo baseline and a lack of nearby feature points, which will be addressed in future work.

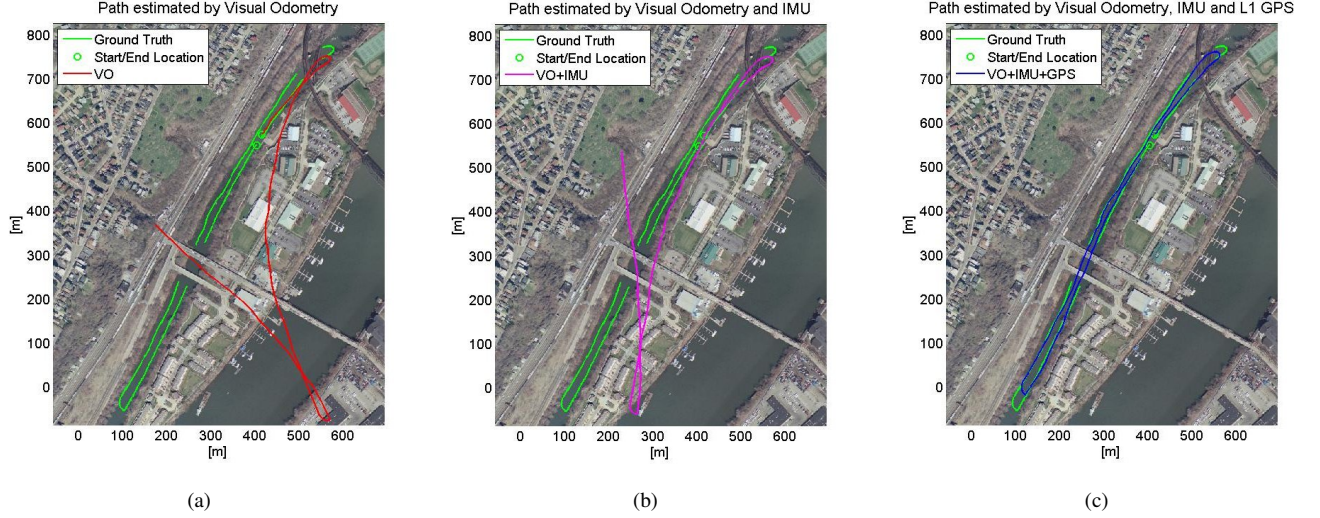


Fig. 9: Estimated path for different sensor suites, overlaid onto an aerial map and ground truth obtained with a highly accurate GPS system. Fig. 9a depicts the estimated path, if visual odometry is used exclusively. Fig. 9b displays the results, if inertial data is incorporated the estimation. In Fig. 9c intermittent position readings of a consumer-grade GPS were used in addition. In Fig. 9a and 9b, the starting positions were picked manually, while for Fig. 9c no information in addition to the sensor inputs was provided.

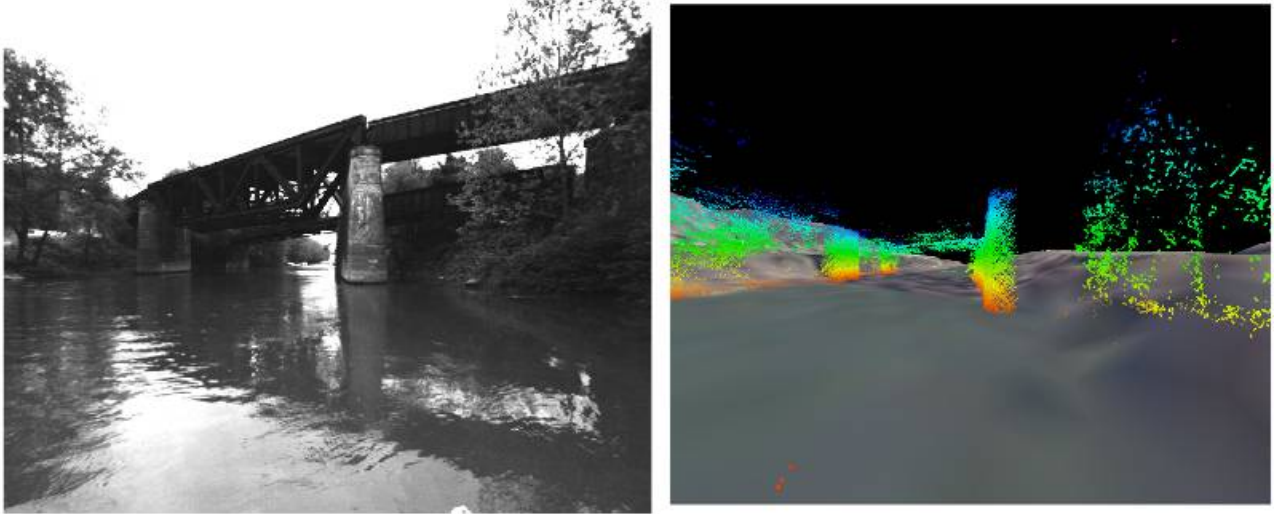


Fig. 10: Example Image and Point Cloud Reconstruction. On the left is an image from the camera. On the right is the reconstructed point cloud built using the pose estimation and overlaid with an aerial image and elevation data.

D. Mapping

Fig. 10 shows examples of the 3D reconstruction built by the laser scanner as the vehicle moves through the environment. Each laser scan is globally registered and placed into a world map by using the filtered state estimate. Since the laser scans occur at a higher frequency than the state estimates, an intermediate state is found by interpolating between neighboring state estimates. The filtered state is locally smooth and accurate enough to build clean 3D reconstructions. The terrain mesh seen in the reconstruction is built from Elevation data and Orthoimagery provided by USGS and the Seamless Data Warehouse [27].

The final desired output of our exploration run is a 3D map describing the structure and a 2D map giving an estimate

of the bank. Fig. 11b shows the map at the end of a 2km run moving up and down a channel with our sensor payload. The map shows that even though GPS data is only integrated sparsely at 0.2 Hz, it is still possible to create a globally registered map of the environment. The individual river classification results are integrated into a map in Fig. 11a. River classification images are calculated at 2Hz and then filtered using an evidence grid into a global representation. The current reliable maximum range for integrating the classification results into the map is about 30 meters. Therefore in some regions, the bank was not visible to the camera.

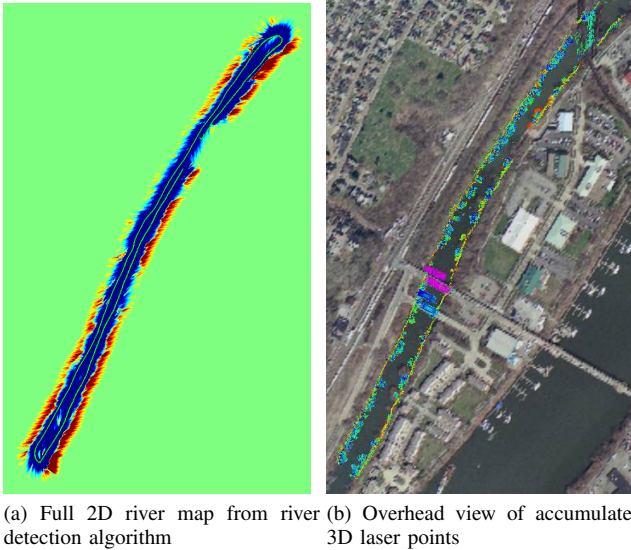


Fig. 11: Maps showing a 2km river traverse (a) River map built from multiple camera images fed into the river detector and fused into an occupancy grid (b) Overhead view of 3D laser point cloud reconstruction globally registered using graph-based state estimator and overlaid on an aerial imagery

VIII. CONCLUSIONS AND FUTURE WORK

We have described the lightweight perception and state estimation systems required to navigate and map a river from a low-flying rotorcraft. We developed an obstacle detection model for our off-axis spinning laser scanner and calculated maximum safe velocities for the rotorcraft. We demonstrated our self-supervised river classification system with improvements for recognizing novel objects. We discussed our graph-based global state estimation and we built a global 2D river map and 3D vegetation map for a 2km river loop.

Our future work will be moving the sensor suite onto our rotorcraft and feeding perception and state information back to a motion planner to realize autonomous exploration and mapping. We are also working to improve scaling issues in visual odometry and incorporate scan matching from the laser scan as an additional input to the state estimator. Finally, satellite imagery or the 3D location of previously identified river segments could provide prior knowledge for the river detection algorithm.

REFERENCES

- [1] J. Leedekerken, M. Fallon, and J. Leonard, "Mapping complex marine environments with autonomous surface craft," in *12th International Symposium on Experimental Robotics*, 2010.
- [2] S. Rathinam, P. Almeida, Z. Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta, "Autonomous searching and tracking of a river using an uav," in *American Control Conference, 2007. ACC '07*, 2007, pp. 359–364.
- [3] J. Yang, D. Rao, S. Chung, and S. Hutchinson, "Monocular vision based navigation in gps-denied riverine environments," in *Proceedings of the AIAA Infotech@Aerospace Conference*, St. Louis, MO, March 2011. [Online]. Available: <https://netfiles.uiuc.edu/sjchung/www/InfotechVisionFinal.pdf>

- [4] A. Rankin, L. Matthies, and A. Huertas, "Daytime water detection by fusing multiple cues for autonomous off-road navigation," in *24th Army Science Conference (ASC'04)*, Nov. 2004.
- [5] A. Rankin and L. Matthies, "Daytime water detection based on color variation," in *IEEE Intl. Conf. on Intelligent Robots and Systems (IROS'10)*, 2010.
- [6] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, "Self-supervised monocular road detection in desert terrain," in *Robotics Science and Systems Conference (RSS'06)*, 2006.
- [7] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," *Robotics Research*, pp. 201–212, 2011.
- [8] E. Fazl-Ersi and J. Tsotsos, "Robot localization in rough terrains: Performance evaluation," in *Computer and Robot Vision (CRV), 2010 Canadian Conference on*. IEEE, 2010, pp. 245–252.
- [9] A. Wu, E. Johnson, and A. Proctor, "Vision-aided inertial navigation for flight control," in *2005 AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2005, pp. 1–13.
- [10] M. Bryson and S. Sukkarieh, "Building a robust implementation of bearing-only inertial slam for a uav," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 113–143, 2007.
- [11] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based slam," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 1036–1049, 2007.
- [12] K. Konolige and M. Agrawal, "Frameslam: From bundle adjustment to real-time visual mapping," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1066–1077, 2008.
- [13] J. Folkesson and H. Christensen, "Graphical slam for outdoor applications," *Journal of Field Robotics*, vol. 24, no. 1, pp. 51–70, 2007.
- [14] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," in *Robotics Science and Systems Conference*, Seattle, USA, June 2009.
- [15] M. Matsumoto and S. Yuta, "3d laser range sensor module with roundly swinging mechanism for fast and wide view range image," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on*, 2010, pp. 156–161.
- [16] N. Fairfield, "Localization, mapping, and planning in 3d environments," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2009.
- [17] A. Folkesson and P. Newman, "High quality 3d laser ranging under general vehicle motion," in *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.
- [18] S. Achar, B. Sankaran, S. Nuske, S. Scherer, and S. Singh, "Self supervised segmentation of river scenes," in *Intl. Conf. on Robotics and Automation (ICRA)*, 2011.
- [19] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.
- [20] A. Geiger, J. Ziegler, and C. Stillner, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.
- [21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," *Journal of Autonomous Robots*, vol. 30, no. 1, pp. 25–39, 2011.
- [22] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [23] M. Lourakis, "Sparse non-linear least squares optimization for geometric vision," *Computer Vision—ECCV 2010*, pp. 43–56, 2010.
- [24] R. Unnikrishnan and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09, July 2005.
- [25] T. Takasu, N. Kubo, and A. Yasuda, "Development, evaluation and application of rtklib: A program library for rtk-gps," in *GPS/GNSS symposium*, 2007.
- [26] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, "A new approach to vision-aided inertial navigation," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, oct. 2010, pp. 4161–4168.
- [27] Seamless data warehouse. [Online]. Available: <http://seamless.usgs.gov/>