The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

3D Reconstruction of Underwater Structures

Chris Beall, Brian J. Lawrence, Viorela Ila and Frank Dellaert

Abstract— **Environmental change is a growing international concern, calling for the regular monitoring, studying and preserving of detailed information about the evolution of underwater ecosystems. For example, fragile coral reefs are exposed to various sources of hazards and potential destruction, and need close observation. Computer vision offers promising technologies to build 3D models of an environment from two-dimensional images. The state of the art techniques have enabled high-quality digital reconstruction of large-scale structures, e.g., buildings and urban environments, but only sparse representations or dense reconstruction of small objects have been obtained from underwater video and still imagery. The application of standard 3D reconstruction methods to challenging underwater environments typically produces unsatisfactory results. Accurate, full camera trajectories are needed to serve as the basis for dense 3D reconstruction. A highly accurate sparse 3D reconstruction is the ideal foundation on which to base subsequent dense reconstruction algorithms. In our application the models are constructed from synchronized high definition videos collected using a wide baseline stereo rig. The rig can be hand-held, attached to a boat, or even to an autonomous underwater vehicle. We solve this problem by employing a smoothing and mapping toolkit developed in our lab specifically for this type of application. The result of our technique is a highly accurate sparse 3D reconstruction of underwater structures such as corals.**

I. Introduction

Motivated by continuing deterioration of underwater ecosystems, and coral reefs in particular, there is a growing interest in adapting techniques such as structure from motion (SFM) and simultaneous localization and mapping (SLAM) to underwater conditions to enable digital reconstruction of these environments. Unfortunately, due to challenging properties of the medium, transferring standard reconstruction methods to underwater environments is a difficult task. Therefore, emphasis has been placed on the reconstruction of sparse distinct terrain features, with the idea that these features are more robust to artifacts from medium effects. As a result, sparse, low resolution models of the seafloor have been obtained using SFM algorithms [1]–[3].

In this paper we propose a technique for large scale sparse reconstruction of underwater structures. The new technique takes stereo image pairs, detects salient features, computes 3D points, and estimates the trajectory of the camera poses. This trajectory along with 3D feature points are used as an initial estimation fed to smoothing and mapping for optimization [4]. The result is an optimized camera trajectory, as well as an optimized 3D point cloud, which in turn is the basis for creating a mesh, which is ultimately textured to obtain a photo-realistic model. These accurate 3D models of underwater environments will enable us to provide ocean scientists with a tool for making quantitative measurements of submerged structures.

Chris Beall, Brian J. Lawrence and Frank Dellaert are with the Georgia Institute of Technology, College of Computing Building, 801 Atlantic Drive,Atlanta, GA 30332, USA. (vila,cbeall3, brianlawrence,frank)@gatech.edu

Viorela Ila is with the Georgia Institute of Technology, Atlanta, GA and also with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain vila@gatech.edu

Viorela Ila has been partially supported by the Spanish Ministry of Science and Innovation under the *Programa Nacional de Movilidad de Recursos Humanos de Investigación*

We gratefully acknowledge the support from the National Science Foundation, Awards 0713162 and 0448111 (CAREER).

II. Related Work

Early techniques for 3D underwater mapping were introduced in [1] which proposed a complete framework for sparse 3D mapping of the seafloor. Problems such as incremental position estimation, recursive global alignment of the final trajectory, and 3D reconstruction of the topographical map were tackled. Other solutions combining SLAM techniques to estimate the position of the cameras with standard 3D reconstruction algorithms allowed mapping of much larger areas. In [5] the vehicle positions are estimated within a visual-based delayed state SLAM framework. The vehicle position estimation incorporates relative pose constraints from image matchings rather than positions of landmarks in the real world. The result is an efficient filtering method for large trajectories to enable accurate reconstruction. This approach was validated experimentally by using monocular imagery on two datasets: a test-tank experiment with ground truth, and a remotely operated vehicle survey of the RMS Titanic. Along the same line, the method proposed in [2] was used within scientific expedition surveys of submerged coral reefs. The result was a composite 3D mesh representation which allowed marine scientists to interact with the data gathered during the mission and to understand the spatial distribution of the large underwater structure. However, poor density reconstruction reduces the possibility of easily identifying the observed structure (corals can hardly be distinguished from rock without referring to field observation).

Pizarro et al. in [6] devoted close attention to low level image processing algorithms, from feature extraction to relative pose transformation between cameras. The result was an enriched SFM technique for sparse 3D reconstruction, where all the steps were revised and adapted to specific underwater conditions. The method was validated within controlled water tank conditions by comparing image based reconstruction to accurate laser scan measurements.

Some steps towards dense underwater 3D reconstruction have been taken in [7], [8]. Dense reconstruction of

submerged structures has been obtained in [7] with the aim to get quantitative measurements of the objects in the scene. But this method worked only in specific experimental setups, where the stereo system was mounted on a controlled manipulator arm, so that the camera rotation and translations were known. By imposing a known trajectory, the number of 3D reconstruction parameters was reduced and a dense recreation of a small object was successfully obtained by applying standard SFM algorithms. However, in large scale applications involving underwater vehicles equipped with vision systems surveying the bottom of the sea within long term missions, such specific restrictions are prohibitive. In [8], after applying a standard SFM algorithm to obtain sparse 3D maps and camera positions, dense depth maps were computed for all pixels in each view.

Present techniques use similar schemes for underwater structure reconstruction; a sparse set of 3D points are first triangulated from visual features and a mesh is generated from the point clouds using Delaunay triangulation. Among all of these earlier methods, variations exist in the way the camera pose is given either by auxiliary position sensors or estimation, and in the way image processing algorithms are adapted to difficult underwater conditions dominated by light attenuation and scattering. For example in [3], from the reconstructed terrain structure, significant surface points with distinct local texture are identified, comprising vertices of a piecewise planar representation of the local surface. As the camera covers new regions of the scene, these terrain features are tracked in subsequent images, new points from these views are added in the same fashion, augmenting the terrain surface model features.

III. Calibration

Accurate calibration of the stereo rig is a crucial first step in creating a high-quality 3D reconstruction. Our stereo rig was calibrated by placing a calibration grid on the ocean floor and recording video of it from various angles. Due to the differing refractive indices between glass and air vs. glass and water, it is best to collect the calibration data underwater. Both cameras were calibrated independently using the Camera Calibration Toolbox for Matlab [9] to obtain the *intrinsic* parameters

$$K = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \tag{1}$$

where f_x and f_y are focal lengths, s is the the skew, which is 0 in our case, and the principal point (c_x, c_y). The *extrinsic* calibration parameters consist of the 3×3 rotation matrix R and the 3×1 translation vector $\mathbf{t}$ which describe the pose of the right camera with respect to the left camera.

Rectification of stereo pairs greatly simplifies the stereo-correspondence problem. The image pairs recorded by the stereo rig are therefore rectified using Bouguet's stereo rectification algorithm [10]. Given the previously determined intrinsic and extrinsic calibration parameters, this method aims to maximize the common viewing areas between the two cameras, while row-aligning the stereo images. The algorithm results in a new camera matrix K, where the left and right camera now share the same focal lengths. A 2D image point (u_L, v_L) in homogeneous coordinates taken from the left camera with an associated disparity d can then be reprojected to 3D coordinates in the left camera coordinate frame by

$$p_i = Q \begin{bmatrix} u_L \\ v_L \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{2}$$

where Q is the reprojection matrix resulting from the rectification, and T_x is the baseline of the stereo rig

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c'_x)/T_x \end{bmatrix} \tag{3}$$

The 3D coordinates with respect to the left camera's coordinate frame are then given by $(x/w, y/w, z/w)$.

IV. Initial Estimation for 3D Point Location and Camera Poses

Our technique computes relative camera pose constraints and locations of 3D points in the environment, which are then used as initial estimate within an optimization process. It takes as input pairs of stereo rectified images obtained from the calibrated stereo rig. The process iterates as follows: Salient image features are extracted and stereo correspondences are established for each pair. The 3D point coordinates are computed for each stereo correspondence in the left camera's coordinate frame. This is discussed with more detail in section IV-A. Next, a set of putatives is obtained by temporally matching features detected in consecutive pairs of images. A 3-point algorithm is employed within a Random Sample Consensus (RANSAC) [11] framework to recover the camera pose rotation and translation for consecutive frames. Details can be found in section IV-B.

A. Feature Detection and Stereo Matching

Robust feature detection and stereo matching are crucial to building a good 3D model. Simple correlation-based features, such as Harris corners [12] or Shi and Tomasi features [13], are commonly used in vision-based SFM and SLAM, ranging from the early uses by Harris himself to the popular work of Davison [14]. These kinds of features can be robustly tracked when camera displacement is small and are tailored to real-time applications. However, given their sensitivity to scale, their matching is prone to fail under larger camera motions; they are even more problematic for loop-closing hypotheses testing. Given their scale and local affine invariance properties, we opt to use SIFT [15] or SURF [16] instead, as they constitute a better option for matching visual features from varying poses. To deal with scale and affine distortions in SIFT, for example, keypoint patches are selected from difference-of-Gaussian images at various scales, for which the dominant gradient orientation and scale are stored. Our

technique produces similar results whether we use SIFT or SURF, with SURF running significantly faster. The results in this paper were generated using only SURF features.

Given the SURF features, we establish matches between left and right images in the usual manner. Specifically, we extract SURF features in both images of a stereo pair, which both generate a 128-element feature descriptor. We then establish stereo matches by computing the Euclidean distance between feature descriptors found in the left and right images. The images are rectified, and as a consequence we are able to restrict the search for stereo correspondences to the epipolar line, and a small region above/below because the calibration may not be perfect. We are mostly interested in tracking features which are within a few meters of the stereo rig, allowing us to further restrict our search window to a certain range of disparities. Restricting the search in this fashion minimizes the computational time expended on stereo matching. The 3D coordinate of each point for which a stereo match was found is computed using the reprojection matrix Q introduced in section III, and stored for later use. The resulting 3D points represent the initial position estimates of the landmarks/vertices in a single frame.

B. Temporal Matching and RANSAC

We track features from frame to frame to recover the structure of the environment. At each iteration, features are matched temporally by individually comparing each feature descriptor from the current pair of images to the feature descriptors in the previous pair of images, restricted to a small search region within the previous image. A linear motion model in image space is used to estimate the position of potential matches. This speeds up the search, and additionally reduces the number of erroneous matches. Putatives are established by computing the Euclidean distance between the 128-element feature descriptors, similar to what was done previously for establishing stereo correspondences.

The incremental rotation R and translation $\mathbf{t}$ which express the current frame's camera pose with respect to the previous one are recovered by way of applying a three point algorithm within a RANSAC framework. SIFT and SURF descriptor matching are quite reliable in many situations, yet RANSAC is needed to eliminate outliers due to erroneous stereo and temporal matching, as outliers are capable of introducing large error into the solution. Composing the incremental rotation and translation for each new stereo rig pose with the previous stereo rig pose yields a camera trajectory in the global coordinate frame. Putative inliers are saved to be used for batch smoothing and mapping in the following stage.

V. 3D Point Cloud and Camera Trajectory Optimization

A. Smoothing and Mapping

Smoothing and Mapping (SAM) is a smoothing (instead of filtering) approach to the SLAM estimation problem [4], [17]. It is a powerful tool based on graphical models which has been efficiently used to estimate the location of a set of landmarks in the environment together with the camera trajectory.

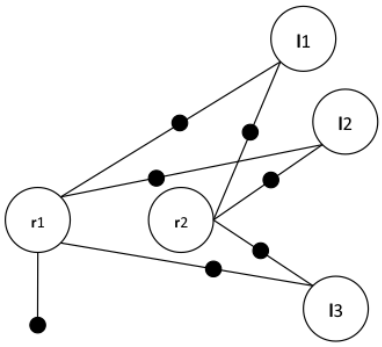

Fig. 1. Factor Graph of two camera poses $\mathbf{r}$ and three landmarks $\mathbf{l}$

SAM enables large-scale mapping, is highly accurate, yet remains efficient in contrast to the state of the art methods based on filtering. While these methods work by estimating only the current state of the vehicle, SAM solves for the entire vehicle trajectory (smoothing), i.e. the position of the underwater vehicle throughout the entire mission. Paradoxically, it was shown in [4], [17] that by asking for *more* (full trajectories) the resulting optimization problems remain *sparse* which results in more efficient mapping algorithms than in the filtering paradigm. In addition, the smoothing approach does not suffer from the consistency issues that are typical in filtering methods: because the full trajectory is always there to re-linearize around, smoothing provides a gold-standard 3D reconstruction that cannot be achieved by any extended Kalman filter approach that "solidifies" linearization errors into the solution.

Factor graphs offer a straightforward and natural way to think of the SAM problem. Fig. 1 shows an illustration of a factor graph which is representative of the optimization problem. The factor graph fully represents the problem to be solved in a graphical way, where the 6 DOF poses of the left stereo rig camera are indicated by $\mathbf{r}$, and the tracked SURF landmarks are denoted as $\mathbf{l}$, where landmarks are represented as 3D points. The nodes on the vertices connecting the variables $\mathbf{r}$ and $\mathbf{l}$ represent the visual image measurements that were made for each tracked feature point.

We optimize over the entire set of camera poses and landmarks, $\mathbf{R}$ and $\mathbf{L}$ respectively, and collect all unknowns in the vector $\Theta \triangleq (\mathbf{R}, \mathbf{L})$. The factor graph then captures a non-linear least-squares problem

$$\Theta^* \triangleq \underset{\Theta}{\operatorname{argmin}} \sum_{m=1}^{M} \|h_m(\mathbf{r}_{i_m}, \mathbf{l}_{j_m}) - \mathbf{z}_m\|^2_{\Sigma_m} \qquad (4)$$

where $h_m(\cdot)$ is the measurement function of landmark l_{j_m} from pose r_{i_m}, and M is the total number of measurements, and $\mathbf{r} \in \mathbf{R}$ and $\mathbf{l} \in \mathbf{L}$ .The measurements are denoted by $\mathbf{z}_m = (u_L, u_R, v)$, where u_L and u_R are the horizontal pixel coordinates, and v the vertical pixel coordinate, all of which result from the projection of a tracked 3D point into the stereo pair. Only one value is needed for v because the

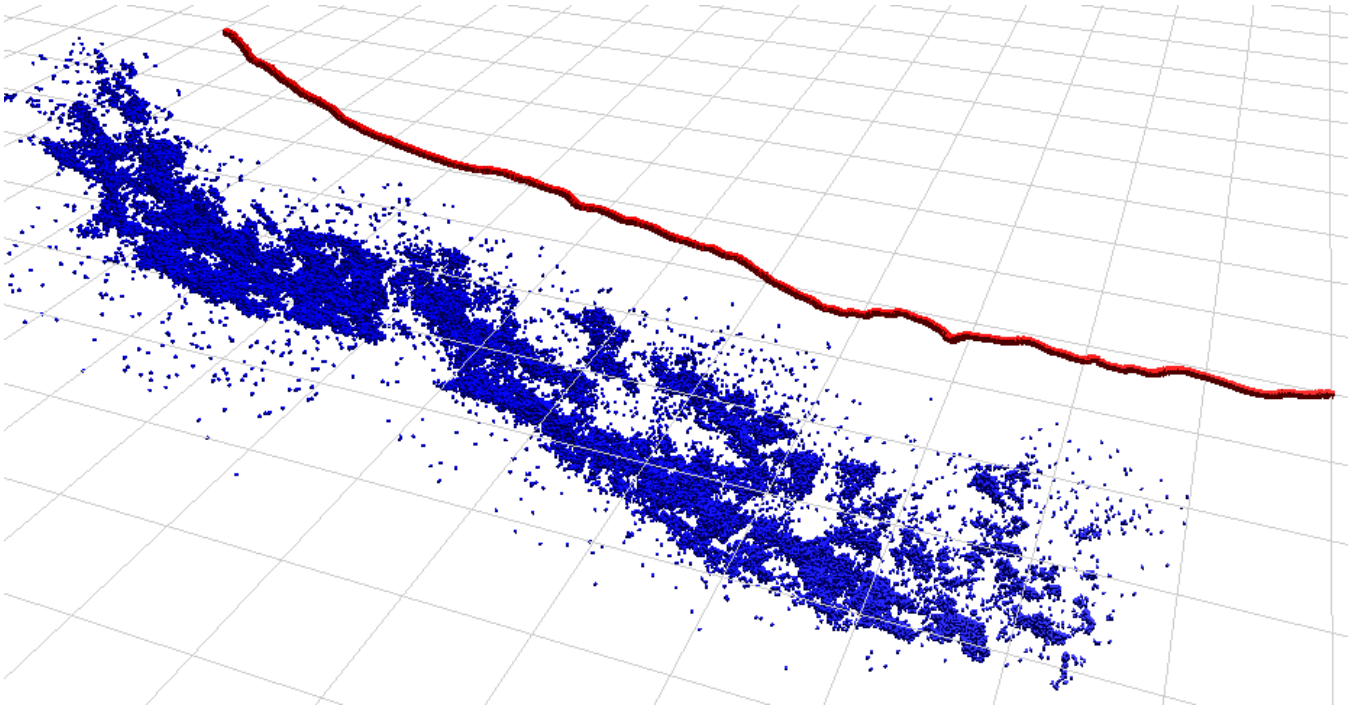

Fig. 2. 670 camera poses and point cloud of 78887 features.

stereo rig is rectified, and hence $v_L = v_R$. In practice one always considers a linearized version of the problem, and Gauss-Newton or Levenberg-Marquardt algorithms are used to solve it. The measurement function $h_m(\cdot)$ takes a point P_i in world coordinates, transforms it into the left camera coordinate frame to obtain p_i, and then projects this point into the stereo pair according to

$$u_L = \frac{fx}{z}, u_R = \frac{f(x - T_x)}{z}, v = \frac{fy}{z} \tag{5}$$

where T_x is the baseline between the stereo cameras. We ultimately solve a standard linearized least-squares problem

$$\delta^* = \underset{\delta}{\operatorname{argmin}} \|A\delta - \mathbf{b}\|^2_{\mathbf{z}} \tag{6}$$

where δ^* is the least squares solution, A results from collecting all of the Jacobian matrices, and $\mathbf{b}$ is the solution of the estimation problem. A is quite sparse and has a typical block structure. Details about the optimization algorithm and its performance can be found in [4]. The result is an optimised 3D point cloud and camera trajectory, which is shown in Fig. 2.

VI. Experimental Results

The data used for this work was collected at coral reefs around Andros Island, Bahamas. Two Canon HV30 consumer HD camcorders were used to form a stereo rig with a wide baseline of $60cm$. Sequences were collected with the stereo rig being guided by a diver, or with the stereo rig strapped to the bottom of a small boat. All of the footage was recorded in 24p mode. The footage was manually synchronized and color corrected before applying our 3D reconstruction technique. The footage has HD resolution of (1920×1080), but nevertheless suffers from the expected effects of underwater light attenuation, motion blur, many moving things such as fish, plants and algae. The speed of the algorithm is correlated to the size of the images, and to the number of features that are being detected. Feature detection is the most time consuming part of the algorithm, limiting the speed of the camera trajectory estimation component to about 1 frame/sec. Fig. 3 shows feature detection and tracking performance by frame. From top to bottom, the graph shows the number of features detected in the left frame, the number of stereo correspondences that were found, and the number of

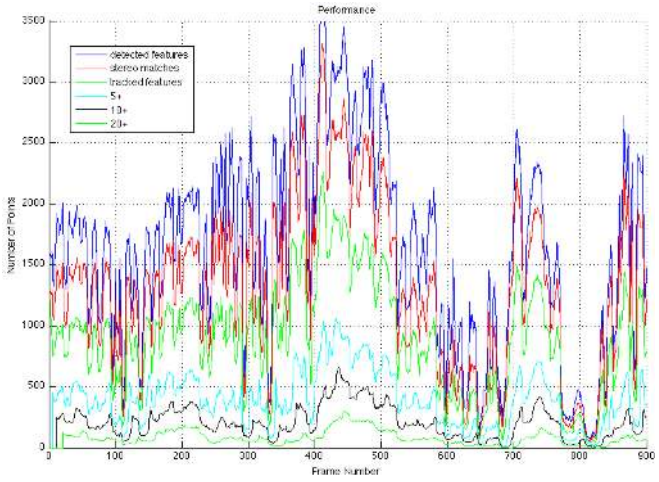

Fig. 3. Feature detection and tracking performance. From top to bottom, the graph shows the number of features detected in the left frame, the number of stereo matches, and the number of features which were tracked for at least one, 5, 10, and 20 frames.

features which were successfully tracked for a minimum of one, 5, 10, and 20 frames. Around frame 800 the number of successfully tracked features drops drastically, and the three-point algorithm is not able to recover the correct incremental camera displacement.

VII. Texture Mapping

To build a realistic 3D reconstruction, a mesh is created from the bundle adjusted point cloud, which is then textured with triangular patches taken from the input video. Depending on where the data was collected, there may be a lot of plants and other protruding irregular shapes on the coral, which are impossible to model accurately in a sparse reconstruction, especially considering that many of these tend to sway in the current. These objects show up as sharp peaks and spikes in the 3D model, and are best ignored in a sparse reconstruction. To address this problem, the bundle adjusted point cloud from the previous step is filtered for spatial outliers using a k-nearest neighbor approach. For each landmark l in the point cloud, the average distance to n nearest neighbors is computed, and a certain percentage of points which have the greatest average distance to their n nearest neighbors is discarded. The percentage of points to discard is manually determined based on the visual appearance of the mesh produced in the next step. The number of plants and roughness of the terrain is positively correlated with this percentage.

A 2D Delaunay triangulation is computed in the $x - y$ plane, as shown in Fig. 4. If too many peaks are apparent at this stage, the k-nearest neighbor filtering in the previous step may need to be adjusted to discard a greater percentage of points.

The final step consists of applying texture to the mesh. To obtain a 3D reconstruction of the highest quality, the texture for each individual face must be sourced from the best frame possible. Naturally, we wish to select textures which have a high resolution, which can be found in the frames where the stereo rig was most nearly frontal parallel and relatively close to the surface to be textured. The camera pose at each frame is already known as a result of bundle adjustment, and surface normals for each face in the mesh are trivial to compute, which enables us to choose the video frame where the stereo rig has the smallest angle of incidence for each face. Textures are copied into square texture maps, sorted by size to make efficient use of the texture maps. An OBJ file is created which contains the representation of the 3D model with references to textures for each face. At this time texture blending is not employed to diminish the appearance of seams. The textured 3D model of the underwater structure is shown in Fig. 5. The model was created by processing 670 frames of video, and contains 63110 vertices. Ground truth is not available, but given the careful camera calibration the model is estimated to be about $7m$ in length. This paper is accompanied by a 3D video animation revolving around the reconstructed surface.

VIII. Conclusions

A highly accurate sparse 3D reconstruction is the ideal foundation on which to base subsequent dense reconstruction algorithms. These algorithms are computationally intensive. By simultaneously estimating a dense reconstruction and the vehicle trajectories, conditioning on pre-built trajectories will yield substantial computational savings. In addition, a *sparse* point cloud is a relative concept: a few tens to hundreds of features per frame will dramatically over-constrain the trajectory estimation, yielding more than sufficient accuracy to serve as the basis for a dense reconstruction.

Having accurate 3D models of underwater environments will enable us to provide ocean scientists with a tool for making quantitative measurements of submerged structures.

References

[1] S. Negahdaripour and H. Madjidi, "Stereovision imaging on submersible platforms for 3-d mapping of benthic habitats and sea-floor structures," *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 4, pp. 625–650, Oct. 2003.

[2] S. Williams, O. Pizarro, M. Johnson-Roberson, I. Mahon, J. Webster, R. Beaman, and T. Bridge, "Auv-assisted surveying of relic reef sites," in *OCEANS 2008. Proceedings of IEEE*, Kobe, Japan, 2008, pp. 1–7.

[3] T. Nicosevici and R. Garcia, "Online robust 3D mapping using structure from motion cues," in *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, April 2008, pp. 1–7.

[4] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Intl. J. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, Dec 2006.

[5] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters for view-based slam," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1100–1114, Dec 2006.

[6] O. Pizarro, R. Eustice, and H. Singh, "Large area 3-d reconstructions from underwater optical surveys," *Oceanic Engineering, IEEE Journal of*, vol. 34, no. 2, pp. 150–169, April 2009.

[7] V. Brandou, A. Allais, M. Perrier, E. Malis, P. Rives, J. Sarrazin, and P. Sarradin, "3D reconstruction of natural underwater scenes using the stereovision system iris," in *OCEANS 2007. Proceedings of IEEE*, Aberdeen, Scotland, 2007, pp. 1–6.

[8] A. Sedlazeck, C. Albrechts, K. Koser, and R. Koch, "3D reconstruction based on underwater video from ROV KIEL 6000 considering underwater imaging conditions," in *OCEANS 2009. Proceedings of IEEE*, Bremen, Germany, May 2009.

[9] J. Bouguet, "Camera calibration toolbox for matlab," 2004. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[10] ——, "The calibration toolbox for Matlab, example 5: Stereo rectification algorithm," 2004. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html

[11] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography," *Commun. Assoc. Comp. Mach.*, vol. 24, pp. 381–395, 1981.

[12] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 147–151, August 1988.

[13] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[14] A. Davison, I. Reid, N. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun 2007.

[15] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," in *Eur. Conf. on Computer Vision (ECCV)*, 2006.

[17] F. Dellaert, "Square Root SAM: Simultaneous location and mapping via square root information smoothing," in *Robotics: Science and Systems (RSS)*, 2005.

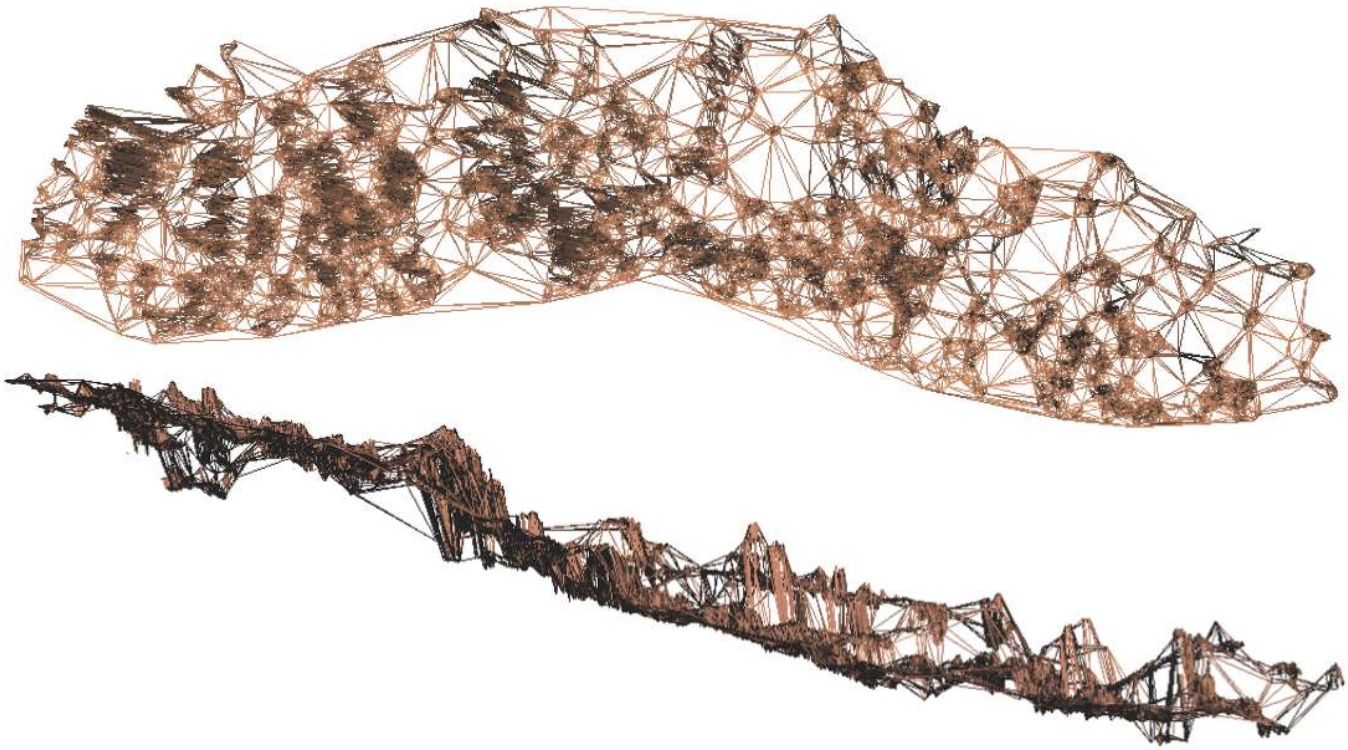

Fig. 4. Delaunay triangulation after outlier removal: top and side view.

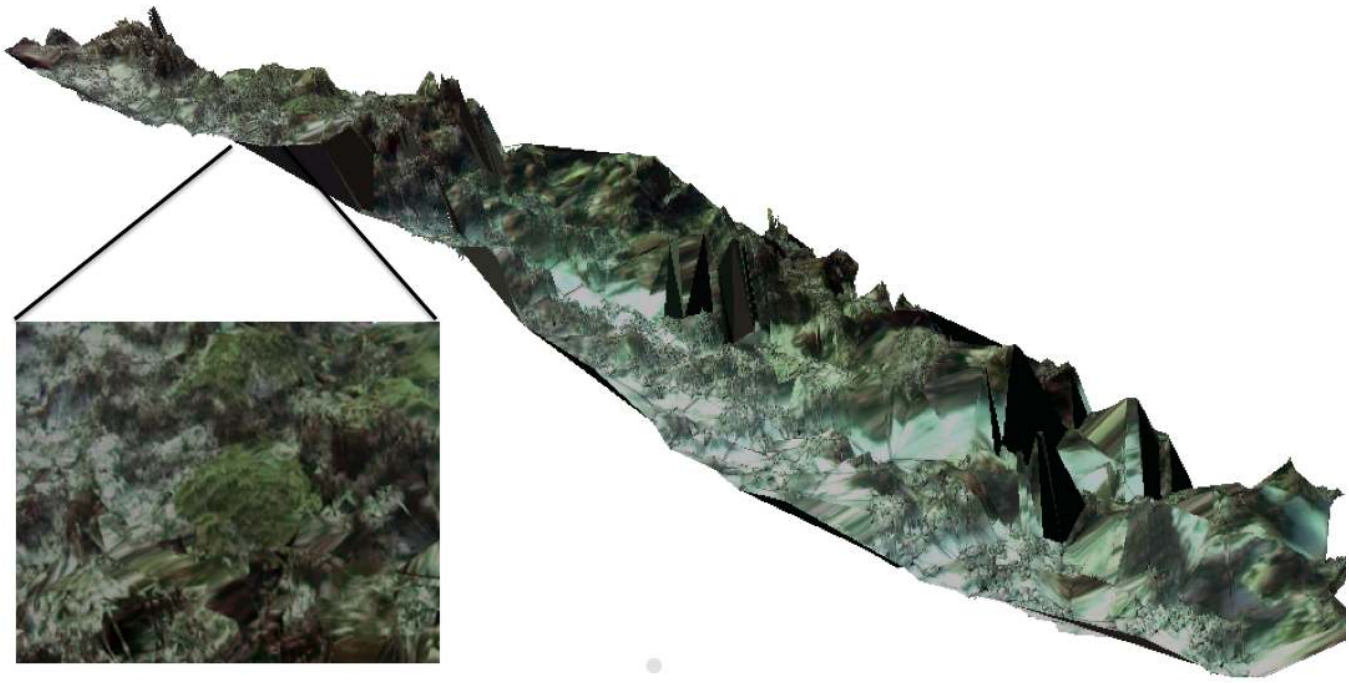

Fig. 5. 3D reconstruction of coral consisting of 670 camera poses and 47thousand faces.