

UNDERWATER CAVE MAPPING AND RECONSTRUCTION USING STEREO VISION

by

Nicholas Weidner

Bachelor of Science
University of South Carolina 2016

Submitted in Partial Fulfillment of the Requirements
for the Degree of Masters of Science in
Computer Science
College of Engineering and Computing
University of South Carolina
2017

Accepted by:

Ioannis Rekleitis, Major Professor

Marco Valtorta, Committee Member

Jason O’Kane, Committee Member

Song Wang, Committee Member

Hossein Haj-Hariri, Dean of the College

© Copyright by Nicholas Weidner, 2017
All Rights Reserved.

ACKNOWLEDGMENTS

This work could not have been completed without the tremendous support of my family, friends, and colleagues. My advisor Ioannis Rekleitis has been an amazing support, and I am grateful for his funding along the way. This work has been all together funded by a Magellan scholarship from the University of South Carolina, the National Science Foundation (NSF 1513203, 1637876), and a Google Faculty Research Award. I am indebted to the cave divers: John Rose, Steve Cox, Mark Garland, and Blake Wilson for collecting the cave video data used for experimentation and producing the results. I am also thankful from the help and support of the other members of my Committee: Marco Valtorta and Jason O’Kane, as well as various members of the Autonomous Field Robotics Lab including Sharmin Rahman, Alberto Quattrini Li, Marios Xanthidis, and Nare Karapetyan.

ABSTRACT

The proposed work presents a systematic approach for 3-D mapping and reconstruction of underwater caves. Exploration of underwater caves is very important for furthering our understanding of hydrogeology, managing efficiently water resources, and advancing our knowledge in marine archaeology. Underwater cave exploration by human divers however, is a tedious, labor intensive, extremely dangerous operation, and requires highly skilled people. As such, it is an excellent fit for robotic technology, which has never before been addressed. The proposed solution employs a stereo camera and a video-light. The approach utilizes the intersection of the cone of video-light with the cave boundaries resulting in the construction of a wire frame outline of the cave. Successive frames produce a scalable accurate point cloud which, through the use of adapted 3-D geometry reconstruction techniques, creates a fully replicated model of the cave system.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	iii
ABSTRACT	iv
LIST OF FIGURES	vii
CHAPTER 1 CAMERA CALIBRATION	5
1.1 Overview	5
1.2 Related Work	6
1.3 Theory	9
1.4 Calibration Data Collection	12
1.5 SuperView	14
1.6 Calibration with MATLAB	14
1.7 Experimental Results	15
1.8 OpenCV Port	18
CHAPTER 2 POINT CLOUD RECONSTRUCTION	21
2.1 Overview	21
2.2 Related Work	21
2.3 Challenges	23
2.4 Wireframe Reconstructon	24

2.5	Visual Odometry	28
2.6	Experimental Results	30
CHAPTER 3 SURFACE RECONSTRUCTION		33
3.1	Overview	33
3.2	Related Work	34
3.3	Algorithm Comparison	36
BIBLIOGRAPHY		43

LIST OF FIGURES

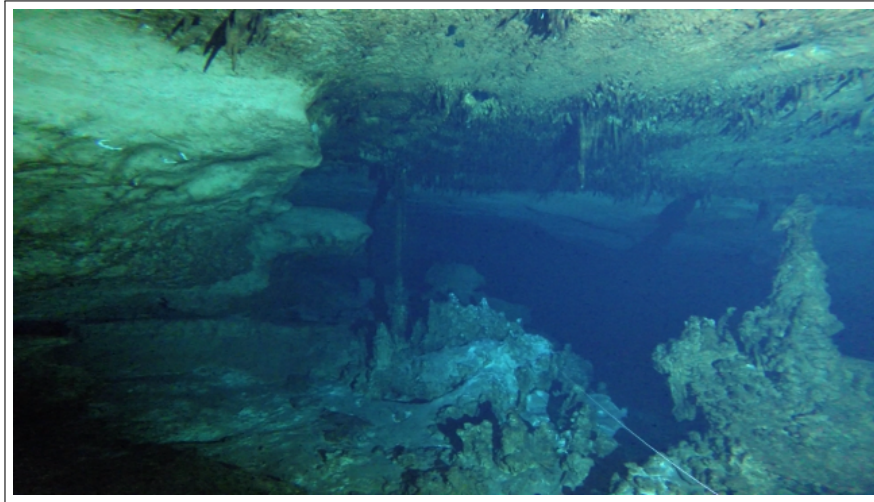
Figure 0.1	0.1a Typical Scene from an underwater cave. 0.1b A cave diver attaching a branch line to the main line of a cave.	2
Figure 1.1	An underwater image of a waterproof calibration checkerboard pattern used for camera calibration.	6
Figure 1.2	Diver calibrating an underwater rig consisting of stereo cameras and an IMU.	8
Figure 1.3	A collection of good viewing angles and distances for a calibration board	13
Figure 1.4	From the same camera pose, (a) 1080p GoPro footage without the SuperView feature. (b) 1080p GoPro footage with the SuperView feature.	14
Figure 1.5	MATLAB Stereo Calibrator App GUI.	16
Figure 1.6	Calibration results using MATLAB, on the left column, results by using all the images, on the right column, results by filtering the dataset using the proposed outlier removal method.	17
Figure 1.7	Initial reprojection errors plotted in python	20
Figure 2.1	Left camera images of an underwater cave with different illuminations. Illumination in the cave is provided by the lights individual divers have and also from a strong video-light. 2.1a Diver in front holds a strong video-light; see how the cone of light outlines the boundaries of the cave. 2.1b Diver with video-light follows behind the camera. 2.1c The diver with the camera also holds the light.	24
Figure 2.2	2.2a Left camera image of an underwater cave. 2.2b The rectified image. 2.2c The rectified image thresholded based on light intensity. 2.2d An edge map of the boundaries of the thresholded image. 2.2e The boundaries filtered to eliminate small contours. 2.2f The longer contours superimposed on the rectified image.	25

Figure 2.3	Select features matched at the boundary between left and right image of a stereo pair.	26
Figure 2.4	Three different reconstructions from two different angles are presented. (a-c) Present a frontal view; (d-f) present a side view. 2.4a, 2.4d Disparity map of the Fig. 2.2b using the the OpenCV's semi-global block matching (SGBM) stereo algorithm. 2.4b, 2.4e Applying the SGBM algorithm only to the lighted part. 2.4c, 2.4f The contour in 3-D using feature matches; see Fig. 2.3. It is worth noting elimination of outliers makes the contours much more distinct.	27
Figure 2.5	ORB features tracked by ORB-SLAM 2.	28
Figure 2.6	2.6a The trajectory calculated by ORB-SLAM 2 of a 7 min 28 sec traversal and the 3-D points estimated from ORB features. 2.6b The wireframe reconstructed from the proposed stereo algorithm. Please note, the reduced number of outliers compared to 2.6a.	29
Figure 2.7	2.7a Average error of the inter-point distance of the target; 2.7b Average error of the reconstructed points from the best plane fitting 3-D points of the checkerboard. The results were from 4,000 images of the calibration target presented to the stereo camera underwater. . . .	31
Figure 2.8	2.8a The 3-D walls of the cave extracted from a short ten second traversal. 2.8b A second view more inside the reconstruction.	32
Figure 3.1	3.1a A point cloud segment of the ten second traversal. 3.1b The same segment with white normal lines protruding inwards	37
Figure 3.2	Results of running Delaunay triangulation.	38
Figure 3.3	3.3a Ball-pivoting applied to the entire point set. 3.3b Applied to a 10,000 sampling. 3.3c Applied to 1,000 point sampling.	40
Figure 3.4	RIMLS marching cubes surface reconstruction	41

INTRODUCTION

The importance of underwater cave mapping spans several fields. First, it is crucial in monitoring and tracking groundwater flows in karstic aquifers. According to Ford and Williams [22] 25% of the world's population relies on karst water resources. Our work is motivated from the Woodville Karst Plain (WKP) which is a geomorphic region that extends from Central Leon County around the "Big Bend" of Florida [47]. Due to the significance of WKP, the Woodville Karst Plain Project (WKPP) has explored more than 34 miles of cave systems in Florida since 1987 [10], proving the cave system to be the longest in USA [29]. This region is an important source of drinking water and is also a sensitive and vulnerable ecosystem. There is much to learn from studying the dynamics of the water flowing through these caves. Volumetric modeling of these caves will give researchers a better perspective about their size, structure, and connectivity. These models have even greater importance than simply enhancing the mapping. Understanding the volume of the conduits and how that volume increases and decreases over space is a critical component to characterizing the volume of flow through the conduit system. Current measurements are limited to point-flow velocities of the cave metering system and a cross-sectional volume at that particular point. My thesis work introduces a first step towards robotic mapping of an underwater cave. The proposed approach results in 3-D reconstructions which will give researchers the above described capabilities. Furthermore, volumetric models will be incredibly helpful for those involved with environmental and agricultural studies throughout the area, and once perfected this technology could help map other subterranean water systems, as well as any 3-D environment that is difficult to map. The Woodville Karst Plain area is sensitive to seawater intrusions which threaten the agriculture and the availability

of drinking water; for more details see the recent work by Zexuan et al. [83]. Second, detailed 3-D representations of underwater caves will provide insights to the hydrogeological processes that formed the caves. Finally, because several cave systems contain historical records dating to the prehistoric times, producing accurate maps will be valuable to underwater archaeologists.



(a)



(b)

Figure 0.1: 0.1a Typical Scene from an underwater cave. 0.1b A cave diver attaching a branch line to the main line of a cave.

Operations in underwater caves can be grouped under three categories: motion inside the known part of the cave; exploration of new territory; and surveying of newly explored areas. Most transportation in the explored part of caves is performed using diver propulsion

vehicles (DPVs). All explored areas are marked by permanently attached cave line, which provides a direct route to the exit; see Fig. 0.1b where a diver is inspecting the line. When divers explore uncharted territory, they proceed without the DPVs, laying out line and tying it to projections on the floor, walls, or ceiling. The third phase, surveying, consists of two divers measuring distances, using a cave-line with knots every 3 m between attachment points. Simultaneously, the divers also measure the water depth at each attachment point, as well as the azimuth of the line leading to the next attachment point. All the information is recorded on a slate or waterproof paper. Estimates of the height and width of the passage can also be recorded, if time permits. The above described process is error-prone and time consuming, and at greater depths results in significant decompression times, where total dive time can reach between 15 to 28 hours per dive. My thesis presents a first step of utilizing robotic technology to assist in cave exploration via the use of a stereo camera and a video-light. In many cases, during DPV rides, the divers attach cameras to their DPV and/or to themselves in order to document the exploration. Consequently, introducing a stereo camera does not complicate the standard operating procedures and will not increase the cognitive load of the divers.

The work of this thesis aims to achieve 3 goals.

1. Study the implementation of camera calibration in a physical setting.
2. Produce a point cloud of the underwater cave utilizing the presence of artificial light.
3. Reconstruct the surface of the cave out of the generated point cloud.

Combined, these pieces will come together to form a sufficient pipeline for reading image data and outputting a mapping of the cave system. This work can be easily adapted to a physical robot setting for later field use as described earlier.

The next chapter will discuss a study of camera calibration along with effective methods for minimizing error in results. Chapter 3 covers the topic of scene reconstruction from stereo vision, and introduces a novel approach to generating point clouds of underwater

cave systems. The last component, point cloud surface reconstruction, will be addressed in chapter 4. Finally, chapter 5 will conclude the work with a discussion of the overall contributions.

CHAPTER 1

CAMERA CALIBRATION

1.1 OVERVIEW

Camera calibration of the data acquisition cameras is a fundamental first step in achieving accurately scaled reconstructions. It is a highly studied and in common setups a solved problem. While much of the research of the community has passed the calibration phase, it is still the fundamental first step in a number of vision based research topics. Accurate calibration models and properly undistorted images are a necessity for any kind of accurate depth measurements, 3D reconstructions, or robotic navigation through a physical space via imaging. Good calibration relies on good input image data and sometimes this can be difficult to obtain. Cameras without digital screens make it impossible to view images or video until connected to a computer. This can make it difficult to validate the calibration set up while gathering input. In the underwater domain, cameras that lack this feature make data validation much more tedious and time consuming as the cameras must be removed from the water and connected to an external machine. If the data is being recorded in the field without additional equipment, it may be impossible to view the images or footage until data collection is complete. This can be even more problematic for stereo calibration systems where it is important that both cameras record the required information. As such, identifying the subset of collected data that results in the best calibration model is an important step in the calibration process.

With such factors as affordability, durability, and quality becoming commonplace for the camera market, the use of cameras in research has grown. One popular brand is GoPro

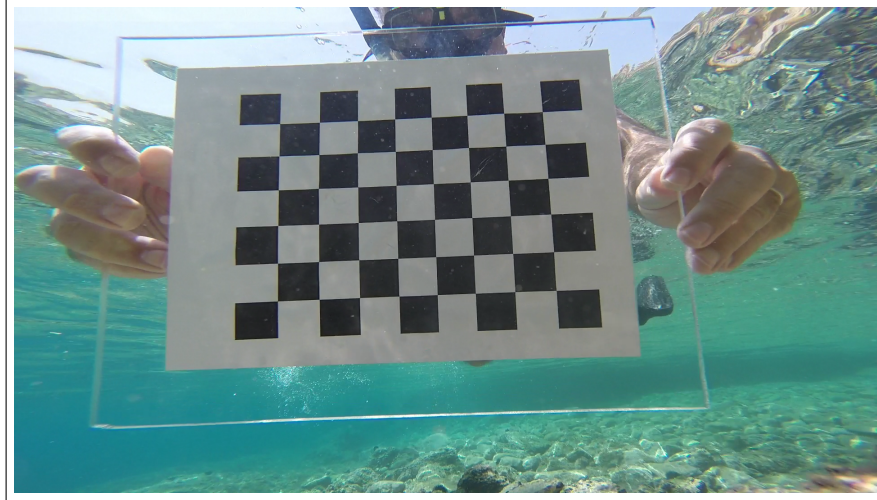


Figure 1.1: An underwater image of a waterproof calibration checkerboard pattern used for camera calibration.

which develops small, durable, high resolution action cameras. GoPro cameras can be used in a number of domains including above and below water and on a number of different applications such as building reconstruction or ocean floor mapping. While the popularity of these cameras has increased, they are susceptible to some of the challenges in collecting good calibration data. Most GoPro models lack a screen for viewing images and footage during and after capture making it hard to position the camera in the scene. Part of this problem has been fixed with the introduction of new Bluetooth and smart phone features, but wireless communication is severely limited in the underwater domain. In the case of this work, there was a need to accurately undistort footage obtained using GoPro's Super-View capture mode. This mode causes severe distortion along the edge of the frame which some calibration tools have trouble overcoming.

1.2 RELATED WORK

Camera calibration is a well established method originating back to as early as the 1900s with lens research by Conrady [13]. This developed into the Brown distortion model which forms the foundation for modern day camera calibration techniques [9, 8, 24]. One of the first openly available camera calibration tools was the Camera Calibration Toolbox for

MATLAB. This tool, developed by Jean-Yves Bouguet, could calibrate a camera and return the intrinsic and extrinsic parameters. The toolbox was built on the foundation of work done by Tsai [80] for introducing off the shelf technologies to camera calibration, Heikkila and Silven [30] for presenting an intrinsic calibration model, and most prominently Zhang for developing many of the techniques used in the toolbox [84]. This work was later ported to OpenCV and used to develop the more powerful Computer Vision System Toolbox for MATLAB. These two tools are widely used in the field today.

More complex calibration models have been developed to handle more complicated camera lenses taking into account different types of distortion. Specific methods to deal with extreme fisheye and barrel distortion have arose both in OpenCV and in independently released packages. Both the Camera Calibration toolbox for MATLAB and OpenCV have a fisheye calibration model based on the work of Kannala and Brandt [39]. Scaramuzza have worked extensively with omnidirectional camera calibration and has developed his own MATLAB toolbox [69, 70, 68, 66]. Currently these are the state of the art readily available tools for rapid prototyping and used by the general public, each with its own advantages and disadvantages.

In recent years, there has been a push for the calibration of low cost and easily portable camera systems. With the development and continued advancement of products such as the GoPro, research has gone into the calibration and use of these cameras to solve imaging problems. Balletti et al. [4] explained many of the advantages of lightweight cameras including their ease to handle, capability of performing under extreme conditions, and providing high quality stills and video. They also explained methods for calibrating the cameras for reconstruction purposes. In the realm of underwater camera calibration, Schmidt and Rzhanov [71], and Shortis [77] compared the results and techniques for calibrating GoPro systems with the added distortion of water. These studies all fail to touch on the calibration of the GoPro camera when it is set to SuperView mode.

Calibration can be much more complicated and tedious if the camera is not the only

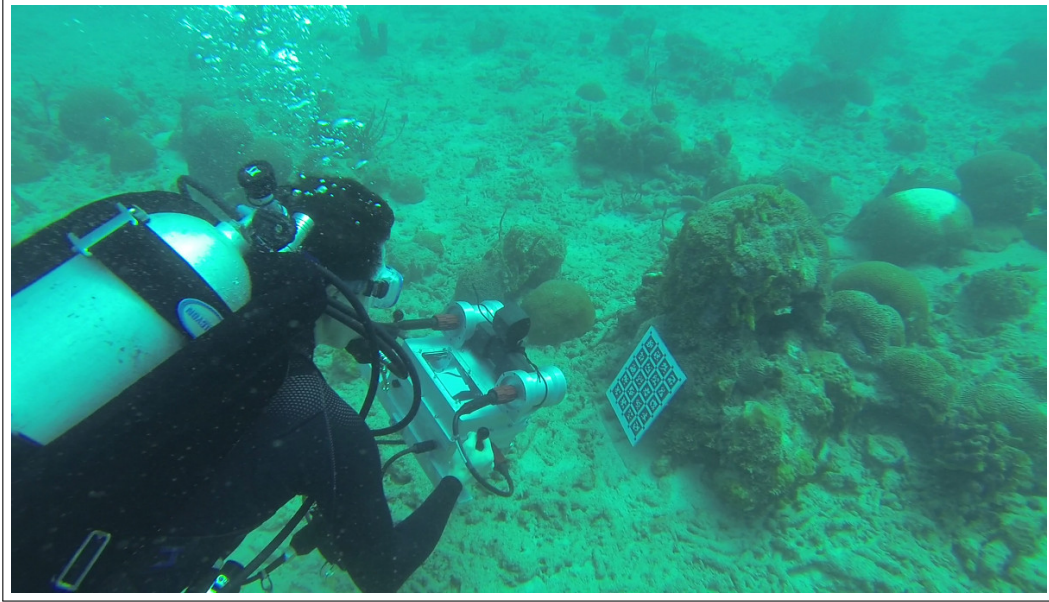


Figure 1.2: Diver calibrating an underwater rig consisting of stereo cameras and an IMU.

sensor needing calibration. Many robotic systems take advantage of multiple proprioceptive and exteroceptive sensors in combination with visual input. A common proprioceptive sensor is the inertial measurement unit (IMU) which measures linear accelerations and angular velocities. Fig. 1.2 shows an example of the calibration process for a visual and interior sensor system. Significant efforts have been in order to calibrate these systems as accurately as possible including the use of a Kalman Filter to determine the unknown coordinate transformations between sensors [56, 52, 41]. Because these calibrations rely heavily on the camera input, it is important that the camera calibration output is as accurate as possible so it will not skew the calibration of the other sensors.

A system for assisting novice users to collect images for calibration through a Graphical User Interface is proposed by Richardson et al. [63].

Hu and Kantor [36] presented a greedy approach for selecting images so that they are uniformly distributed over different camera poses. Such a method considers a budget that encodes the maximum processing time allowed. A quality metric could be considered to improve the performance.

1.3 THEORY

The pin hole camera model is the simplest model for how a camera works and is used as the basis for modern implementations of camera calibration. With this model, intrinsic and extrinsic camera parameters define the transformation between 3D world coordinates and 2D image coordinates. The intrinsic parameters disregard the position and orientation of the camera and define such parameters as the focal length, principal point, and aspect ratio. Extrinsic parameters define the pose between object frame and camera frame in terms of rotation and translation. The full model to get the pixel coordinates (u, v) of a point in the world (X, Y, Z) , including intrinsic and extrinsic parameters, is

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1.1)$$

where λ defines the scale, u and v describe the coordinates of the newly projected image pixel point, f_x and f_y describe the focal length values in pixels over width and height of the camera sensor, c_x and c_y describe the principal point coordinates in pixels, r_{ij} represent the elements of the extrinsic rotation matrix, t_i represent the elements of the extrinsic translation matrix, and X , Y , and Z are the 3D coordinates in the world reference frame. The extrinsic matrix is what translates 3D coordinates X , Y , and Z to new coordinates X_{new} , Y_{new} , and Z_{new} in the camera reference frame. This transformation equates to

$$\begin{bmatrix} X_{new} \\ Y_{new} \\ Z_{new} \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \quad (1.2)$$

Then, using triangle equivalence, assuming a unit distance between the Center of Projection where all the ray lights project in the camera and the image plane, the pixel X'_{new} and Y'_{new}

in normalized image coordinates can be determined as follows, :

$$X'_{new} = X_{new}/Z_{new} \quad (1.3)$$

$$Y'_{new} = Y_{new}/Z_{new} \quad (1.4)$$

Taking into account the intrinsic parameters of the camera, we can then find the projected u and v pixel coordinates on the actual image plane:

$$u = f_x * X'_{new} + c_x \quad (1.5)$$

$$v = f_y * Y'_{new} + c_y \quad (1.6)$$

When actual cameras are examined, calibration must take into account the distortion resulting from the lens. Distortion is separated between radial and tangential distortion. Radial distortion is caused by the light bending near the edges of the lens. MATLAB defines the distortion along the x (columns) and y (rows) coordinates of the camera as:

$$x_{rdistortion} = X'_{new}(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (1.7)$$

$$y_{rdistortion} = Y'_{new}(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (1.8)$$

where k_i are the radial distortion coefficients and $r^2 = X'^2_{new} + Y'^2_{new}$. Tangential distortion is the result of the lens unparallel to the camera sensor plane. MATLAB defines the x and y of this distortion as

$$x_{tdistortion} = [2 * p_1 * X_{new} * Y_{new} + p_2 * (r^2 + 2 * X'^2_{new})] \quad (1.9)$$

$$y_{tdistortion} = [p_1 * (r^2 + 2 * Y'^2_{new}) + 2 * p_2 * X'_{new} * Y'_{new}] \quad (1.10)$$

where the p_j are the tangential distortion coefficients and $r^2 = X'^2_{new} + Y'^2_{new}$. These distortions can be introduced into the perspective projection as

$$u = f_x * X'_{new} * (x_{rdistortion} + x_{tdistortion}) + c_x \quad (1.11)$$

$$v = f_y * Y'_{new} * (y_{rdistortion} + y_{tdistortion}) + c_y \quad (1.12)$$

resulting in the pixel coordinates u and v in the image plane.

Using known world coordinates and identifying how the camera transforms points in a 2D image, the intrinsic $[f_x, f_y, c_x, c_y]$ and extrinsic $[R|t]$ parameters can be calculated, where R and t are the matrix and vector that represents the rotation with elements r_{ij} and the translation with elements t_i . Testing the accuracy of the resulting model is done by reprojecting the points backwards through the model and identifying how close the points lie to where they were originally. The average distance these points are off from where they should be is the reprojection error and is used to identify how well the model is. Reprojection error measures pixel units and ideal models should minimize reprojection error as much as possible.

Once the error is at an acceptable level—typically below 1 pixel—the images can be undistorted using the calibration parameters. Because the intrinsic parameters have nothing to do with the image scene, they can be used to undistort any collection of images taken with the same camera. If the domain changes, for example the camera is moved from above water to underwater, the distortion coefficients will no longer accurately represent the image distortion. Because of this, the camera needs to be calibrated for both underwater and on above water undistortion. While theoretically the introduction of a water distortion model could eliminate the need for calibrating directly underwater, practically this was found unnecessary.

While this describes the process of monocular camera calibration, in a stereo system there is also the step of rectification. Rectified images are undistorted images with an additional rotation applied in order to make the epipolar lines of the image parallel. They also use an altered projection matrix that shifts the images apart by a distance described by the baseline of the cameras. The baseline is the distance between the optical centers of each camera. In a horizontal stereo setting, matching epipolar lines in each camera would have identical y coordinates. The new rectified projection matrices in a horizontal stereo

system are

$$P1 = \begin{bmatrix} f_x & 0 & c_{x1} & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.13)$$

$$P2 = \begin{bmatrix} f_x & 0 & c_{x2} & T_x * f_x \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1.14)$$

where T_x is the horizontal shift amount. The rectified u' and v' coordinates can again be defined as

$$u' = P1 * R * P1^{-1} * u \quad (1.15)$$

$$v' = P2 * R * P2^{-1} * v \quad (1.16)$$

and the rotation matrix R is defined so the epipolar lines along both images are now parallel. This is a vital step in any kind of reconstruction work, because it allows a matching algorithm to reduce the search space to find the distance between matching features in stereo image sets; the output can then be used for the triangulation of 3D world points using such a distance, also called disparity.

1.4 CALIBRATION DATA COLLECTION

The common approach among calibration techniques is to use a set of points with known relationships, for example points in a grid pattern. The points are usually extracted from a set of input images as the corners, or centers of blobs. On these input images distinct features are needed so that it is possible to easily identify the spatial relationship of points in space compared to how the camera perceives them. Common approaches use calibration targets with black and white squares or circles with known dimensions. The checkerboard pattern is one of the standardized approaches, see Fig. 1.1. The intersection of white and black squares creates distinct points in space, all the points exit on the same plane along

parallel lines, and the square size provides accurate measures physical distance between the points. Thus, the undistorted (rectified) images can be checked if they preserve the linear relations between points.



Figure 1.3: A collection of good viewing angles and distances for a calibration board

When collecting images using a calibration target there are a number of important factors to consider. The input images should show the calibration board at a variety of locations, depths, and angles. Fig. 1.3 shows a collection of images taken when calibrating the stereo rig shown in Fig. 1.2. The calibration was performed using the OpenCV fisheye camera calibration functions. This set of images highlights the important lessons learned. First there are views of the board from different distances, from far to near as seen left to right respectively. Second, the board is skewed with respect to the image plane; this is achieved by tilting the target to be non parallel to the camera. Third, the board should never be rotated past 90 degrees otherwise the images can be flipped; it is worth noting that using an odd by even dimension calibration pattern makes the board robust to orientation changes. The pattern should be visible in the field of view in its entirety, although this is an obvious point, when there are no preview capabilities it can be challenging. In particular, when calibrating a stereo system the pattern should be fully visible in both the left and the right camera.

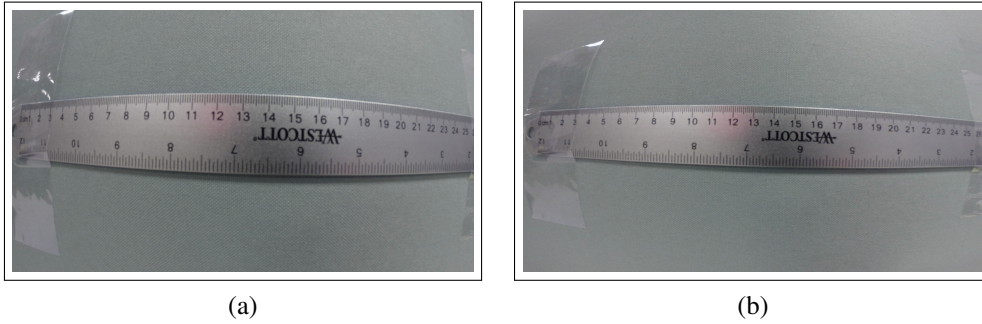


Figure 1.4: From the same camera pose, (a) 1080p GoPro footage without the SuperView feature. (b) 1080p GoPro footage with the SuperView feature.

1.5 SUPERVIEW

The GoPro cameras starting with Hero 3+ are employing a video capture mode termed *SuperView*. According to GoPro, SuperView works by dynamically stretching the 4:3 aspect ratio of the original image into a 16:9 aspect ratio. This keeps the center of the frame unchanged, but severely distorts the edges. The outcome is an image that captures more of the standard 1080p resolution mode. The exact image transform GoPro uses to fit the new aspect ratio is undisclosed so removing the distortion in post processing is not an option. As can be seen in Fig. 1.4, there is a drastic difference in images collected in GoPro’s SuperView mode and their scene vertically and adjusts the aspect ratio to fit a standard wide screen. Many standard video editing software packages such as Adobe Premier, Adobe After Effects, or GoPro’s own GoPro Studio have undistortion features, but these apply a stylistic look which does not help to calibrate the camera properly.

1.6 CALIBRATION WITH MATLAB

As mentioned earlier, there are two standard, widely used, packages for camera calibration, one in MATLAB and one in OpenCV; both based on the same theoretical formulation. The MATLAB Computer Vision system Toolbox calibration implementation includes a feature for visualizing the reprojection error across input images as a graph; see Fig. 1.5. This

graph also plots a mean reprojection error line which can help to identify which input images caused the most problems with the calibration. There is also an adjustable threshold line to set a maximum acceptable reprojection error, resulting in removing outlier images with reprojection error above the set threshold. In other words, by setting this threshold level after calibration, the system can reject the images above the line and rerun the calibration on only those images below. If the input data set is large (several tens to hundred of images), this can help cut down the input images to a more manageable number between twenty and fifty. Furthermore, when calibrating, there is an option to use two or three distortion coefficients. The GoPro cameras we experimented with, calibrate best using two coefficients, but cameras with severe radial distortion work better with three.

While adjusting such features as the number of coefficients or the reprojection error threshold can result in a better camera model, it is important to avoid false positive reprojection error results. Using too many distortion coefficients can minimize the reprojection error, but rectified images generated with those parameters can be dramatically distorted. It is also important when removing outliers to avoid fitting the model to a non diverse image set. When removing outliers, try to keep images from distinct calibration pattern positions with lowest reprojection error. Also viewing the rectified images and scene reconstructions can validate if the parameters were accurate or not. Careful examination of the outliers images captured in SuperView mode indicated that points detected near the boundary of the image were badly misplaced due to the extreme distortion.

1.7 EXPERIMENTAL RESULTS

Experimental Setup In our results we used a 8x6 calibration board with 25mm squares when underwater, and a 7x6 calibration board with 29mm squares when on air. We found the use of an odd by even size calibration board avoided the possible problem of upside down corner detection. In our underwater footage this did not interfere with our underwater results so our old set up still used. The board was printed on waterproof paper and glued

onto an acrylic board as smoothly as possible to avoid any extra noise.

Calibration Procedure and Results Using the left and right frames extracted from our calibration video, the images were run through the MATLAB Computer Vision system Toolbox Stereo Camera Calibrator upon which a result was determined. The reprojection error graph was examined in order to identify image pairs that exceeded the average image pair mean. These pairs were then removed from the input image list and the process was rerun. This happened repeatedly until either the total reprojection error fell within a specified reasonable range, or the image pair set became too small for meaningful calculations to be done.

Using this method, we were able to calibrate a GoPro camera recording in SuperView mode with reprojection error of 0.9 on air and 0.51 underwater, compared to an error of 2.5 on air and 1.5 underwater without any outlier removal, as seen in Fig. 1.6.

Fig. 1.6 compares the reprojection error results of image sets both with and without SuperView and both above and below water. In each case, calibration parameters were determined using a subset of full input data set—training phase—where the outliers had

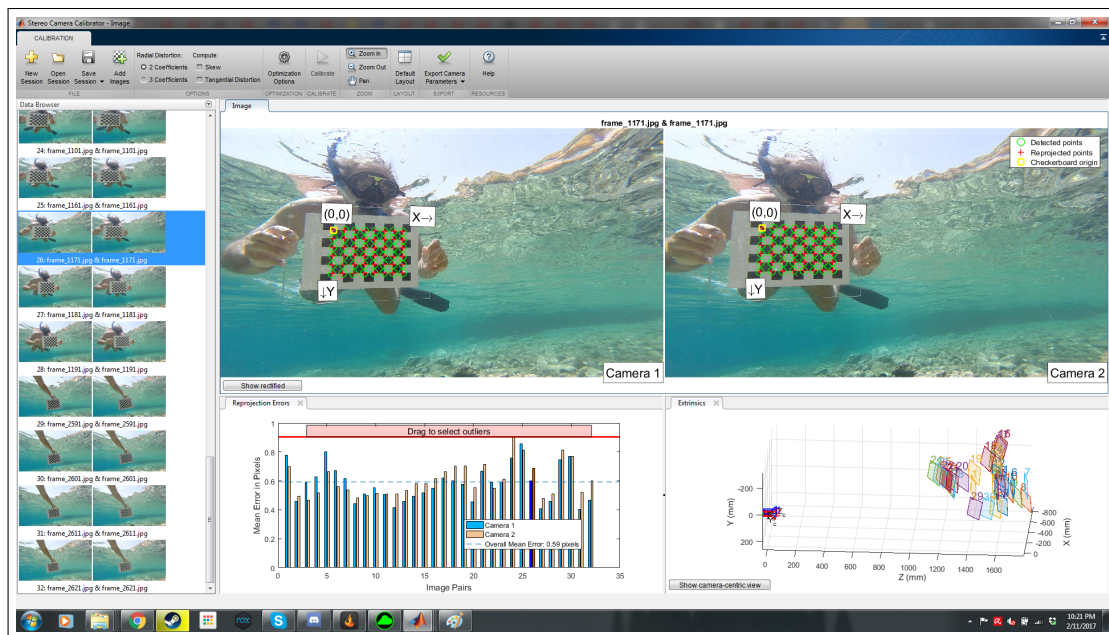
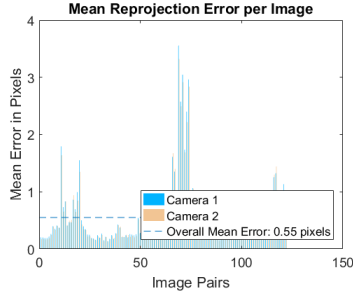
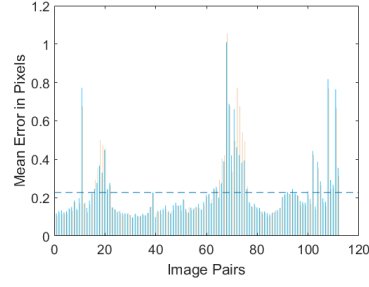


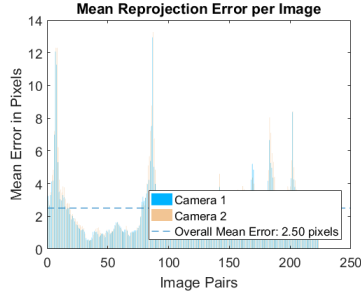
Figure 1.5: MATLAB Stereo Calibrator App GUI.



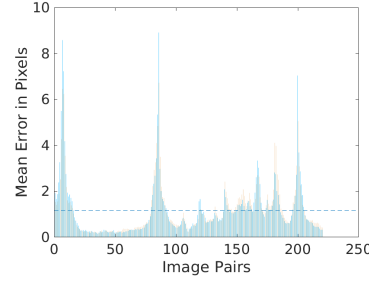
(a) Non-filtered underwater dataset without SuperView



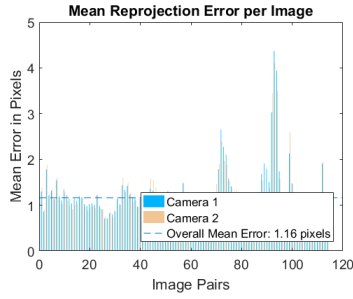
(b) Filtered underwater dataset without SuperView



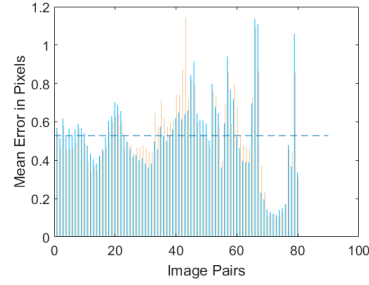
(c) Non-filtered underwater dataset with SuperView



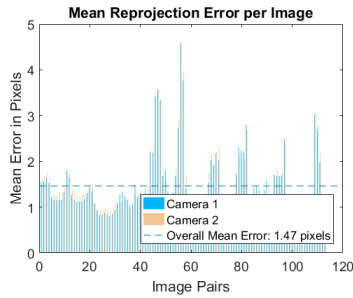
(d) Filtered underwater dataset with SuperView



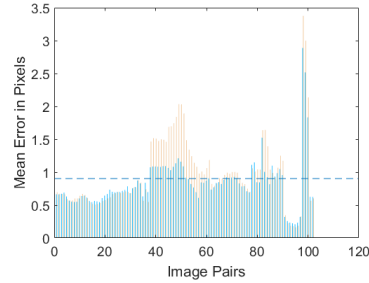
(e) Non-filtered above water dataset without SuperView



(f) Filtered above water dataset without SuperView



(g) Non-filtered above water dataset with SuperView



(h) Filtered above water dataset with SuperView

Figure 1.6: Calibration results using MATLAB, on the left column, results by using all the images, on the right column, results by filtering the dataset using the proposed outlier removal method.

been removed. These parameters were then used to calculate the reprojection error of the full set once again—validation phase. Note that, in MATLAB, the function provided that recovers the extrinsics given the set of 2D-3D points of the checkerboard is based on a closed form formulation, while the one used inside the function for estimating the camera parameters is using a numerical optimization algorithm. As such, reprojection errors calculated on the same images from the training phase and the validation phase could be slightly different. In our experiments it was around 1-2 pixels. As such, we formulated the recovery of the extrinsics as a numerical optimization problem and we compute the error using a numerical optimization algorithm. It can be observed that the mean pixel error is lower on the full set when the parameters from the image subset are used. Many plots still have outlier cases which result from a few possibilities. One cause is that while the corners were all detected by the corner detector, there was motion blur that skewed the actual location of the points. A second cause is the SuperView dynamic stretching that pulls the corner points outward near the edges. This effect is not completely eliminated after calibration which signifies features very close to the edge of the scene are not usable in any future processes. A future study might evaluate the region for which undistorted SuperView images remain undetected by the dynamic stretching. Comparing the results of image sets above and below water, underwater sets had a number of more sharp outlier cases. This is most likely the results of the difficulty in actually acquiring underwater calibration footage. The two domains did both had success in producing low reprojection error. In our case we even got slightly better error results from using SuperView underwater compared to above water, possibly because of the additional water distortion effect.

1.8 OPENCV PORT

While the MATLAB Computer Vision system Toolbox is highly powerful and produced good results for our setup, there are drawbacks. Namely MATLAB is close source software that requires payment to use. At the same time, MATLAB can be tedious to integrate with

external applications and thus might not be ideal if camera calibration is only a part of a larger pipeline. OpenCV on the other hand is both widely used for a variety of image and computer visions applications, and is both open source and easy integrate. As noted earlier, OpenCV has implementations of camera calibration already and they are simple to use. Unfortunately they lack the user interface and built in outlier removal to make calibrating something like SuperView efficient. A similar implementation of the MATLAB outlier removal system was thus developed as an easy to use tool in OpenCV.

Because data visualization plays such a crucial role in identifying bad calibration input images, there needed to be a good way to generate interactive graphs of the OpenCV calibration results. This can be time consuming in C++, so alternatively python's Matplotlib library was used to recreate as similar output as possible to MATLAB's. OpenCV has both a C++ and a python library, but python was found to quite slow in calibrating large data sets. Due to this, a hybrid C++ and python application was created. The finalized control flow is as follows.

1. The C++ application reads in a set of input image pairs for calibration
2. The calibration board corner points are detected and stored in an external file
3. The first calibration run takes place and the reprojection error of every input image is stored in an external file along with the average reprojection error
4. A python script reads the reprojection errors and plots them along with an average reprojection error line
5. The user adjusts the average reprojection error line, selects outlier removal button, and the file of corner points is edited down to the new smaller set
6. The C++ application reads in the corner points, avoiding time spent calculating the corner points again, and recalculates the reprojection errors with the new set
7. The output is again used in the python script and the process is repeated

Fig. 1.7 shows the plot generated by the OpenCV port, and the reprojection error for the image set of 1.5. After iterative outlier removals, the error is brought down from the original 2.83 to 1.02. This compares to MATLAB's error output of .51. Because MATLAB is a closed source application, the exact method for calibration can not be ported over directly. This means MATLAB's calibration probably involves a slightly refined calculation and optimization. OpenCV has trouble matching MATLAB's results and its speed, but compared, but this method still provides an open source alternative. These results match or are even better than hand picking input images and saves time eliminating bad options.

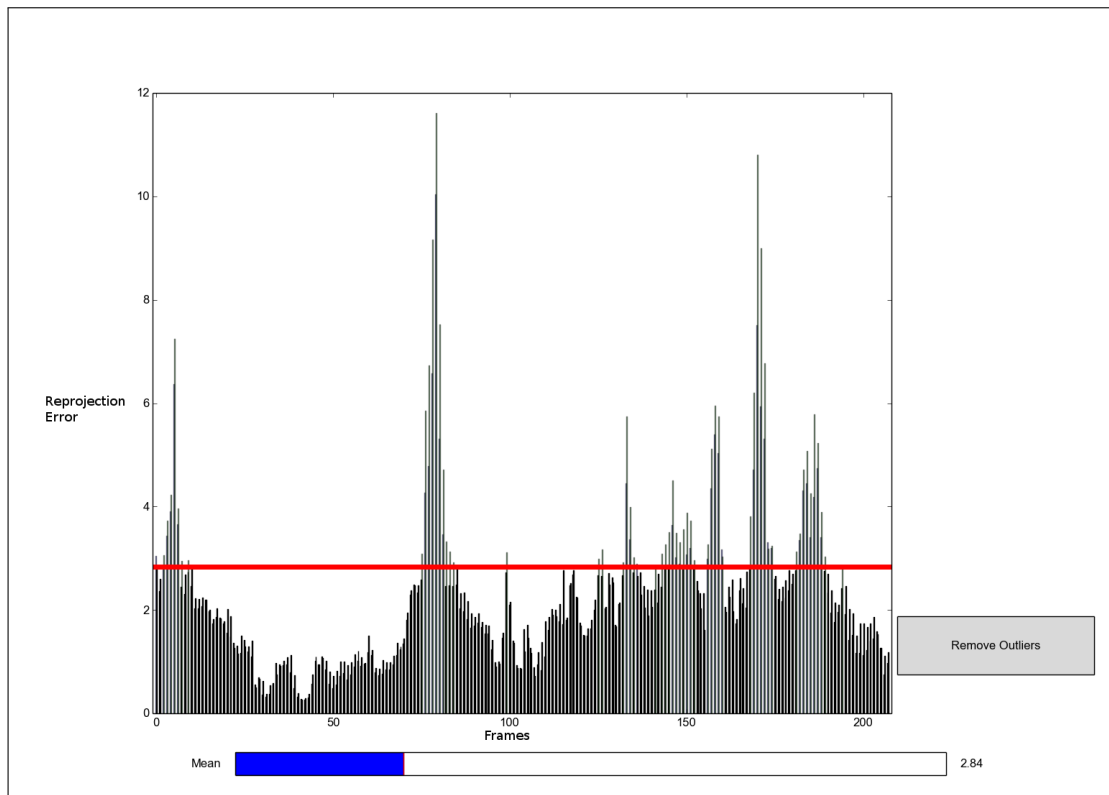


Figure 1.7: Initial reprojection errors plotted in python

CHAPTER 2

POINT CLOUD RECONSTRUCTION

2.1 OVERVIEW

Camera calibration was an important step in our overall pipeline of underwater cave reconstruction, because without it we could not accurately project 2-D image points into 3-D space. The footage for all of our tests were taken using a GoPro in SuperView mode, so this calibration problem needed to be addressed. With calibration out of the way, we now shift focus to the generation of a geometrically accurate point cloud using stereo footage collected in an underwater cave. The presented approach utilizes the presence of the artificial lighting to produce a rough model of the traversed area. In particular, the video-light cone is used to identify the walls of the cave from a single stereo pair. Furthermore, motion between consecutive stereo pairs is estimated and the 3-D reconstruction is utilized to produce an approximate volumetric map of the cave.

2.2 RELATED WORK

The majority of underwater mapping up to now consists of fly-overs with downward pointing sensors mapping the floor surface. The resulting representation consists of 2.5 dimensional mesh-maps or image mosaics with minimal structure in the third dimension. In addition to underwater caves, several other underwater environments exhibit prominent three dimensional structure. Shipwrecks, are significant historical sites. Producing accurate photorealistic 3-D models of these wrecks will assist in historical studies and also monitor their deterioration over time. During maritime disasters, it is important to produce accurate

maps of the sunken vessel, especially the interior, in order to assist with rescue efforts. Multiple cases existed where survivors have been rescued from submerged vessels [26] hours, or even days after the event. Finally, underwater infrastructure inspection [62] is another dangerous and tedious task that is required to be performed at regular intervals. Such infrastructure includes bridges, hydroelectric dams [64], water supply systems [81], and oil rigs. For more information please refer to the Massot-Campos and Oliver-Codina survey [55] for an overview of 3-D sensing underwater.

Most of the underwater navigation algorithms [48, 78, 37, 65] are based on acoustic sensors such as Doppler Velocity Log (DVL), ultra-short baseline (USBL) and sonar. Gary et. al. [25] presented a 3D model of a cenote using LIDAR and sonar data collected by DEPTHX (DEep Phreatic THERmal eXplorer) vehicle having DVL, IMU and depth sensor for underwater navigation. Corke et. al. [14] compared acoustic and visual methods for underwater localization. However, collecting data using DVL, sonar, and USBL while diving is expensive and sometimes not suitable in cave environments.

Using stereo vision underwater has been proposed by several groups, however, most of the work has focused on open areas with natural lighting, or artificial light that completely illuminates the field of view. Small area dense reconstruction of a lit area was proposed by Brandou et al. [7]. Mahon et. al. [54] proposed a SLAM algorithm based on the viewpoint augmented navigation (VAN) using stereo vision and DVL in underwater environment. A framework proposed by Leone et al. [49] operated over mainly flat surfaces. Several research groups have investigated the mapping and/or inspection of a ship's hull using different techniques [32, 35, 20, 43], the most famous shipwreck visual survey being that of the Titanic [21]. Error analysis was performed recently by Sedlazeck and Koch [73]. The problem of varying illumination was addressed by Nalpantidis et al. [59] for above-ground scenes in stereo reconstruction. More recently, Servos, Smart and Waslander [74] presented a stereo SLAM algorithm with refraction correction in order to address the transitions between water, plastic, and air that exist in the underwater domain.

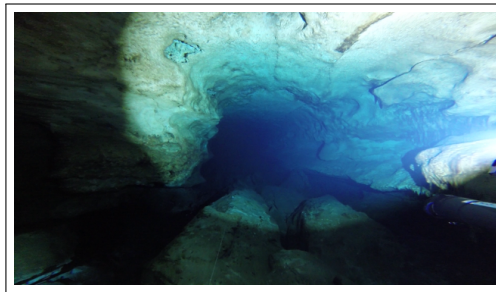
2.3 CHALLENGES

As can be seen in Fig. 2.1, the complete absence of natural illumination in combination with the presence of several sources of artificial illumination, such as: each diver’s primary light and also one or more video-lights, results in huge lighting variations in the scene. In particular each diver’s primary light generates a tightly focused beam which is constantly moving with the motion of the diver. In Fig. 2.2a, there are three divers present: one holding the video light, his tanks visible at the bottom of the image; one traveling with the camera, not visible; and a third one whose DPV is visible at the top of the image. The primary light of the third diver can be seen as a blue beam pointing downwards, starting at the left of the DPV.

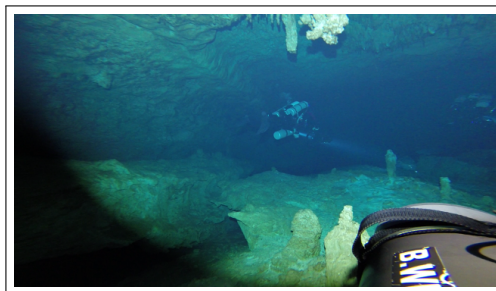
The lighting variations make the success of traditional visual odometry [60] algorithms near impossible. The main assumption of Brightness Constancy Constraint underlying most visual odometry algorithms is violated by the constantly moving light-sources. Table 2.1 presents tests of five open source packages of vision based SLAM on underwater cave vision datasets; as expected most of them failed on the longer sequence and the rest were not able to extract the scale of the environment. It is worth noting that several of the packages are expecting specific motions in order to initialize [44]. Complete results are not presented due to space constraints; interested readers should refer to the work of Quattrini Li et al. [51] for a detailed analysis of more packages and a variety of datasets. The selection of the algorithms presented here was motivated of testing a variety of methods; feature based [57, 45], semi-direct [23], direct [19], and global [72]. The main challenge these algorithms face is the constant change of the field of view and the dramatic lighting variations resulting from occlusions and from light absorption over distance. Among the most successful was the ORB-SLAM [57] with its latest incarnation as ORB-SLAM 2, still in beta version, working with stereo images. While some of these packages, produced an acceptable trajectory, their shape reconstruction from the detected 3-D features was plagued by noise.



(a)



(b)



(c)

Figure 2.1: Left camera images of an underwater cave with different illuminations. Illumination in the cave is provided by the lights individual divers have and also from a strong video-light. 2.1a Diver in front holds a strong video-light; see how the cone of light outlines the boundaries of the cave. 2.1b Diver with video-light follows behind the camera. 2.1c The diver with the camera also holds the light.

2.4 WIREFRAME RECONSTRUCITON

Using light variations to infer shape has been used extensively in the past [42, 18, 46]. The 3-D reconstruction consists of several steps. First the images have to be rectified; a process achieved through a process called camera calibration.

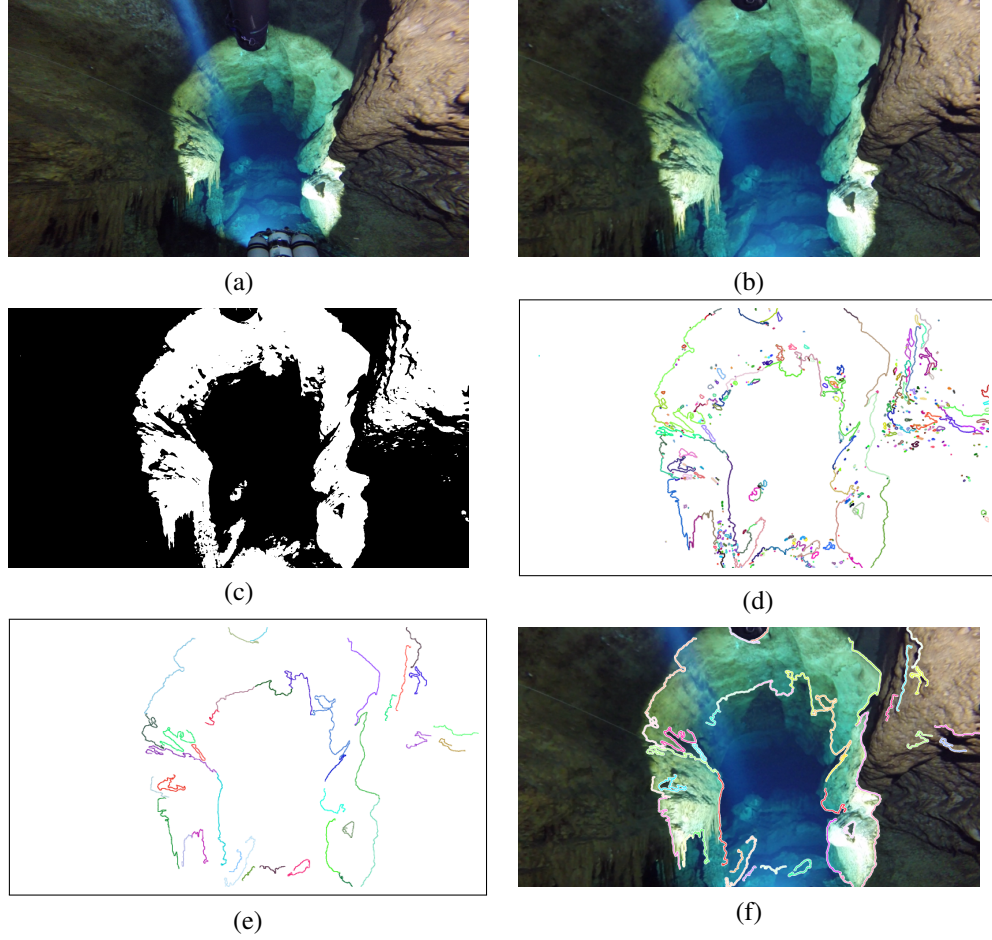


Figure 2.2: 2.2a Left camera image of an underwater cave. 2.2b The rectified image. 2.2c The rectified image thresholded based on light intensity. 2.2d An edge map of the boundaries of the thresholded image. 2.2e The boundaries filtered to eliminate small contours. 2.2f The longer contours superimposed on the rectified image.

Table 2.1: Performance of different open source vision based SLAM packages on underwater data; for a detailed analysis please refer to Quattrini Li et al. [51]

	[57] ORB-SLAM	[45] PTAM	[23] SVO	[19] LSD-SLAM	[72] Colmap
10 sec	noisy	no initialization	partial trajectories/no scale	loss of track	partial trajectories/no scale
448 sec	noisy	no initialization	partial trajectories/no scale	loss of track	partial trajectories/no scale

Contour Tracking Adaptive thresholding is used in order to identify the areas with different illumination; see Fig. 2.2c for the thresholded image where the cone of the video-light meets the cave walls. Selecting the right value for thresholding the image required some domain knowledge, and currently was performed per video sequence, by a human. Cur-

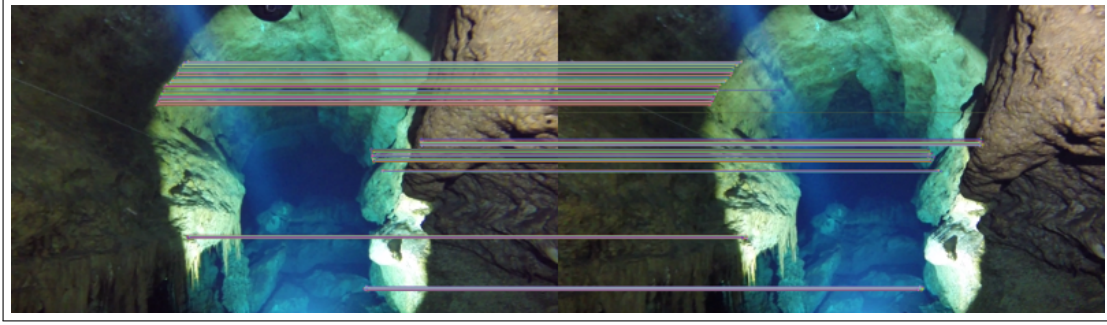


Figure 2.3: Select features matched at the boundary between left and right image of a stereo pair.

rent experiments consider adjusting the threshold based on keeping a balance between the amount of light and dark areas, but that work is outside the scope of this paper.

During the next step, edge detection marks the boundaries between light and dark areas; see Fig. 2.2d for the boundaries of Fig. 2.2b. The OpenCV Canny edge detector [12] is used to identify the edges marking the lighter area boundaries. As can be seen, the edge map is very noisy and thus not suitable for estimating the walls of the cave. A filter is applied to the contour list, eliminating short contours. More specifically, for every contour, its bounding box is calculated and then only the highest fifth percentile is kept. While this method can eliminate elongated contours, experiments with the actual underwater cave video footage proved to not affect the main boundaries. The filtered contours can be seen in Fig. 2.2e. Figure 2.2f superimposes the filtered contours on the rectified image; the areas where the cone of light meets the cave walls are clearly identifiable. In addition, the area with acceptable lighting is extracted for use at the motion estimation. The edge map of the boundaries is used then as input to a stereo reconstruction algorithm.

Sparse Stereo Reconstruction The 3-D structure of the cave boundaries is estimated for each stereo pair. For every point on the contour of the left image a SURF feature descriptor [5] is calculated using the left rectified image. Consequently, the same descriptor is matched on the right rectified image. Outlier rejection is facilitated by searching only locations at the same row and to the right of the left-image feature’s coordinates. As the camera

calibration error is 0.8 pixels, it justifies the assumption above. Previous work on feature quality [75, 76, 82] for underwater images indicated SURF [5] to be the most appropriate feature descriptor. Furthermore, the OpenCV Canny edge detector groups the edges in a list of continuous contours, as such consecutive points belonging to the same contour can be filtered for consistency. Figure 2.3 presents select feature matches corresponding to the contours between the left and right image of a stereo pair.

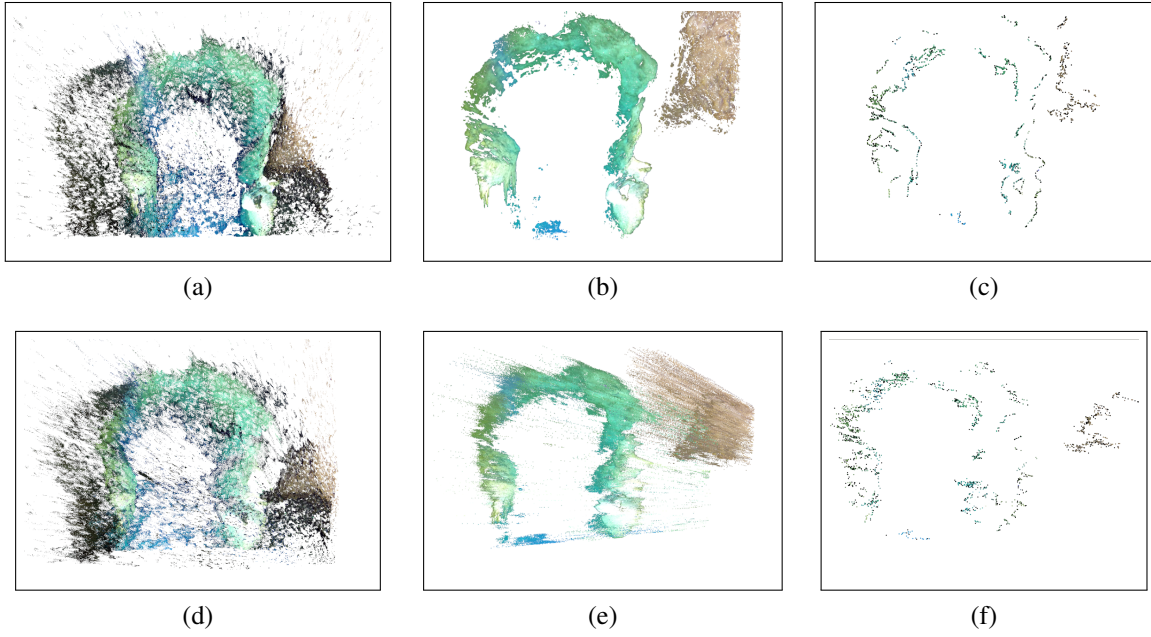


Figure 2.4: Three different reconstructions from two different angles are presented. (a-c) Present a frontal view; (d-f) present a side view. 2.4a, 2.4d Disparity map of the Fig. 2.2b using the the OpenCV's semi-global block matching (SGBM) stereo algorithm. 2.4b, 2.4e Applying the SGBM algorithm only to the lighted part. 2.4c, 2.4f The contour in 3-D using feature matches; see Fig. 2.3. It is worth noting elimination of outliers makes the contours much more distinct.

Figure 2.4 presents a comparison of the performance of dense stereo reconstruction using OpenCV's semi-global block matching (SGBM) stereo algorithm [31] and the contour calculation. The standard output of dense stereo algorithms is a depth map, a normalized image where depth is quantified between 0 and 255; as such the values are discretized; see Figs. 2.4a, 2.4d for the 3-D reconstruction using the SGBM stereo algorithm on Fig. 2.2b.

The noise is quite noticeable, Figs. 2.4b,2.4e present the same reconstruction using only the lighted areas. Finally, Figs. 2.4c, 2.4f present only the contours of high intensity variation extracted from Fig. 2.2b projected in 3-D using SURF feature matching between left and right image. The noise is largely reduced, and the cave boundaries are clearly identifiable. While the first row of Fig. 2.4 presents a frontal view, and the error is not noticeable, the second row, presents a side view and the outliers are obvious. Currently, the corresponding points are calculated with pixel accuracy which results in disparity estimates that are integers. Consequently, the depth estimation is discretized and as it is inversely proportional to the distance of the camera there is a scattering effect. While this effect is strong during dense reconstructions, see Fig. 2.4c, it is also present on the contour estimation; see Fig. 2.4d

2.5 VISUAL ODOMETRY

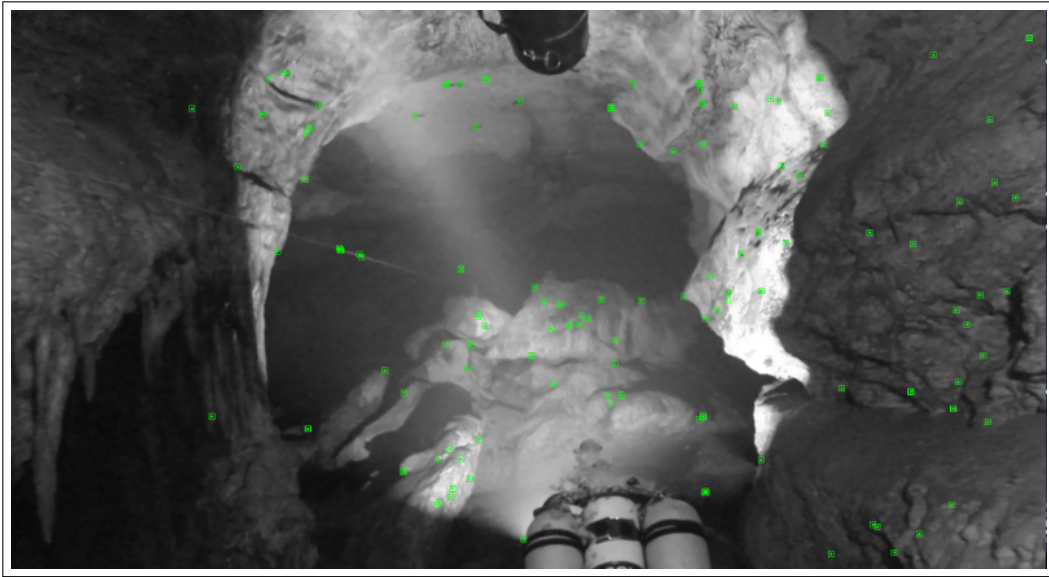


Figure 2.5: ORB features tracked by ORB-SLAM 2.

Brute force application of VO algorithms [17, 15] is quite challenging in the underwater cave domain, due to the lighting variations and the sharp contrasts existing in the image, as discussed earlier. However, by thresholding the image, the areas of adequate illumination in the left and right camera feed can be used to apply one of the latest VO algorithms,

ORB-SLAM 2, a variant of ORB-SLAM [57, 58] for stereo vision. Figure 2.5 presents tracked features in the areas with higher illumination. It is worth noting that during some segments of the video the third diver swimming below the camera exhaled sending a cloud of bubbles in the field of view, however, the VO algorithm was robust enough to handle these dynamic features. This event highlighted one of the challenges of underwater vision.

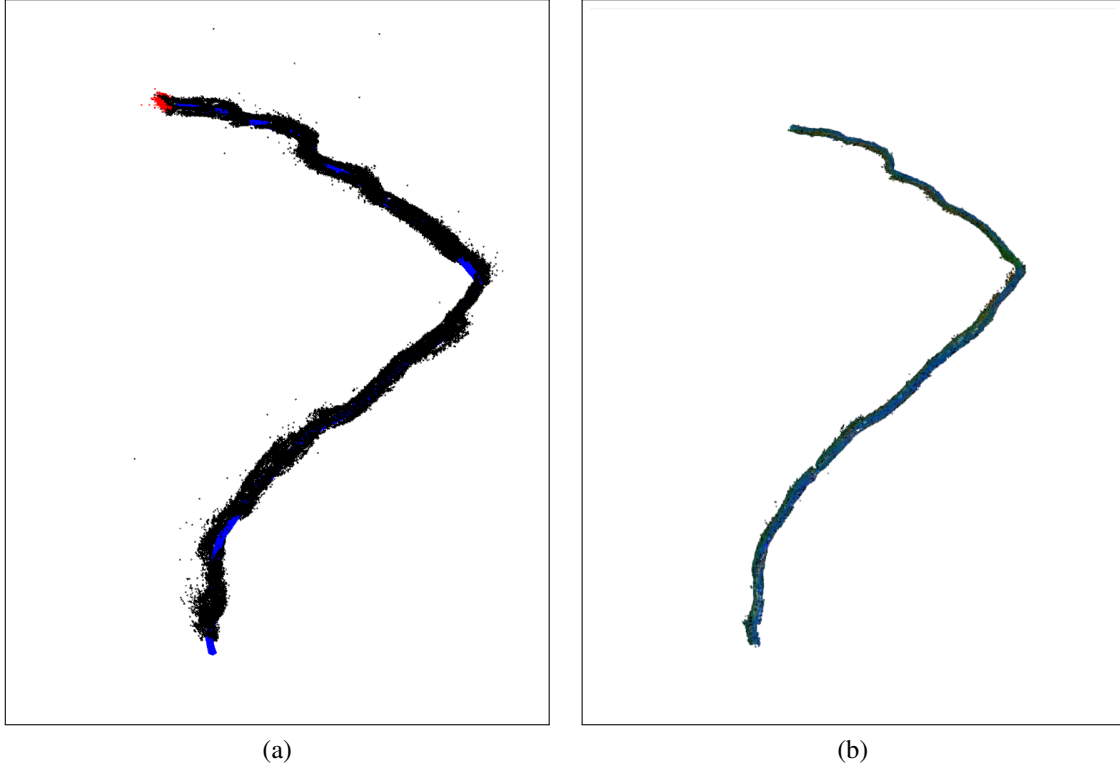


Figure 2.6: 2.6a The trajectory calculated by ORB-SLAM 2 of a 7 min 28 sec traversal and the 3-D points estimated from ORB features. 2.6b The wireframe reconstructed from the proposed stereo algorithm. Please note, the reduced number of outliers compared to 2.6a.

Figure 2.6 presents the trajectory of the stereo camera and the 3-D position of stable features as extracted from ORB-SLAM 2 from a trajectory of seven minutes, twenty eight seconds. While there was no ground truth, observing the video one gets a qualitative verification for the estimated trajectory. The estimated trajectory is then used as an input to produce a volumetric map by transforming the boundaries calculated above through space using the estimated pose of the stereo camera at each instant. It is clear that the contour

based reconstruction; see Fig. 2.6a, has eliminated several outliers which were present in the ORB-SLAM reconstruction; see Fig. 2.6b. The next section presents results from an actual cave.

2.6 EXPERIMENTAL RESULTS

In January 2015, the authors requested from a cave exploration team in Mexico to acquire sample footage using a Dual Hero stereo camera from GoPro during a dive at an already explored cave. The selected cave is part of the Sistema Camilo, the 11th longest submerged cave system in the world, located at Quintana Roo, Yucatan peninsula, Mexico. The camera was mounted on a DPV and the video-light was carried in different configurations in order to demonstrate alternative lighting schemes.

Camera Calibration As mentioned above, the stereo camera used utilizes a recording mode termed superview, which stretches the image in order to produce more aesthetically pleasing videos. Post-processing all the calibration footage collected, error analysis showed, as expected, the error to slightly increase with distance; see Fig. 2.7.

Stereo Reconstruction Figure 2.6b presents the 3-D reconstruction of a long video of 7 min 28 sec. The structure corresponds with the cave morphology, however it is difficult to discern in the still image. Figure 2.8 presents the 3-D reconstruction of a cave segment from a short ten seconds traversal. The left and right walls are clearly identifiable, while the floor and ceiling are occluded from the two divers that swam in the field of view.

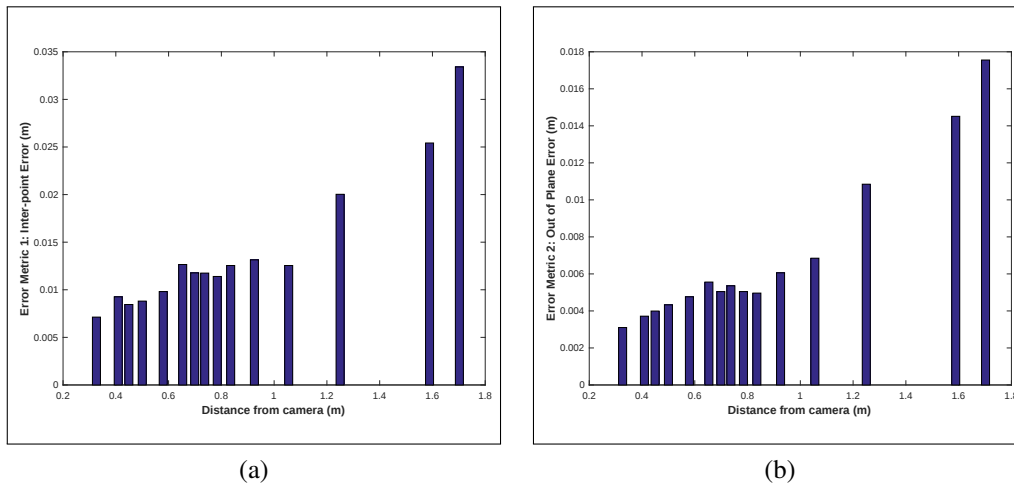
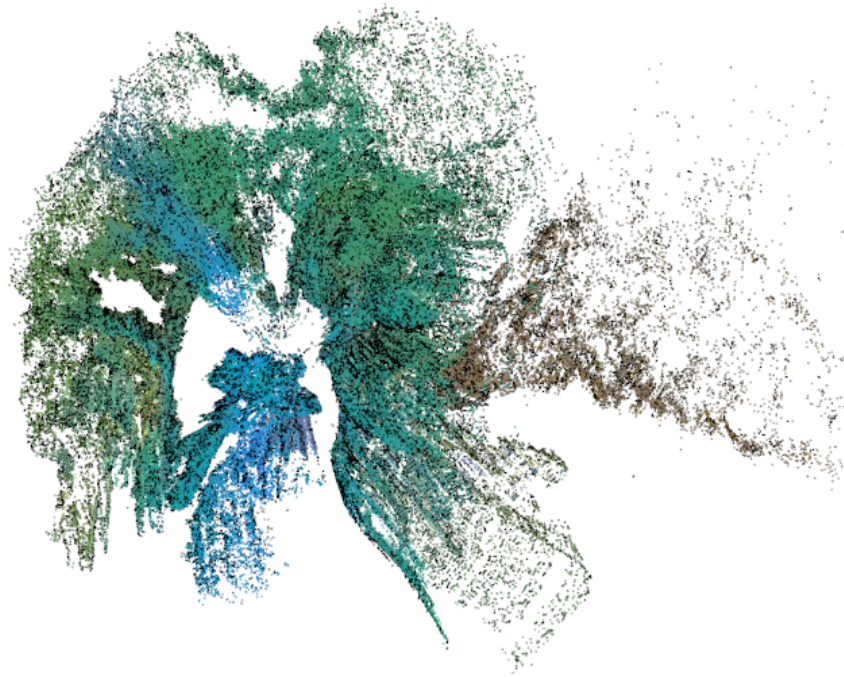
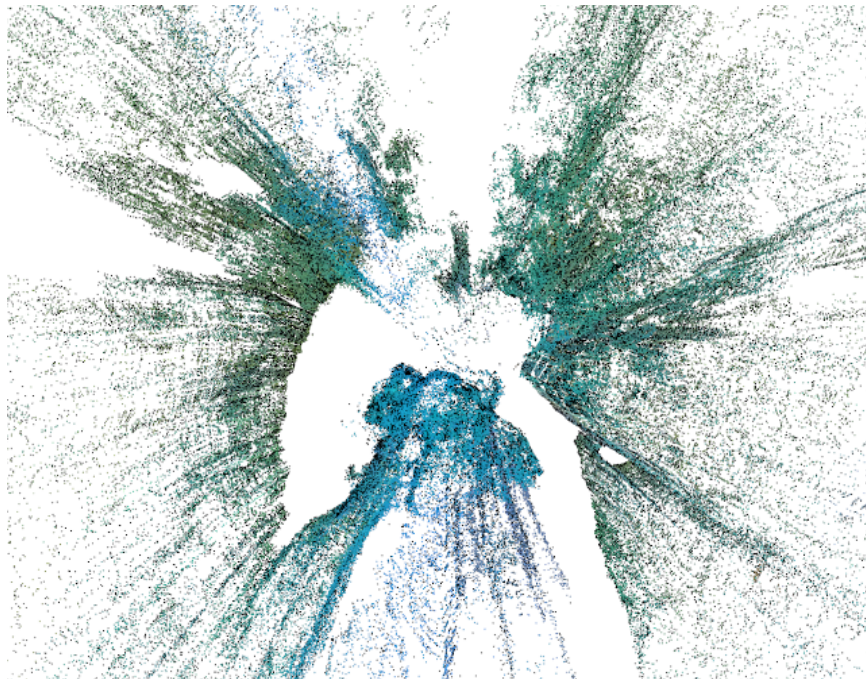


Figure 2.7: 2.7a Average error of the inter-point distance of the target; 2.7b Average error of the reconstructed points from the best plane fitting 3-D points of the checkerboard. The results were from 4,000 images of the calibration target presented to the stereo camera underwater.



(a)



(b)

Figure 2.8: 2.8a The 3-D walls of the cave extracted from a short ten second traversal. 2.8b A second view more inside the reconstruction.

CHAPTER 3

SURFACE RECONSTRUCTION

3.1 OVERVIEW

A geometrically sound point cloud gives a lot of information about the 3-D structure of cave. The shape can be easily inferred, and features of the cave such as stalagmites take shape. It is also easy to see how the cave system is shaped on a large scale when longer segments are constructed such as in 2.6. What a point cloud lacks though is information between all of the points. This space can be either large or small depending on the how many image frames were used to build the model. These empty spaces can cause visual problems if shapes loose their definition due to lack of information. If these models were to be used for robot integration as discussed earlier, the area of these holes cannot be used for any kind of sensing information. Due to this need for additional information in our reconstructed model, surface reconstruction is used in order to fill the empty spaces. The goal is a reconstruction of a full model that continuously defines the shape of the cave structure and overlays texture information to give visual definition to the shape. Our reconstruction problem is not unique enough to warrant a brand new surface reconstruction technique. As will be explained in section 3.2, a lot of work has gone into the problem of surface reconstruction. We hope to identify the best method out there for our particular problem, and build upon those results to refine our final output even more.

3.2 RELATED WORK

Surface reconstruction is a problem that expands well beyond scene reconstruction from a camera. Many methods have developed for data collected with range scan datasets. Luckily these methods can often be applied generically to any kind of data set, but there are key components and limiting factors that make some better than others. Such data sets as those collected with underwater stereo images often fail under the constraints of popularized techniques.

A subcategory of surface reconstruction techniques is that of point interpolation. In this category, the input data points would all represent points exactly on the surface of the reconstruction object. Because this is in practice impossible without at least some minimal error, applications take measurement error into account, but assume it to be small. These methods are thus popular on range scan data sets where the volume of the data points is high, the surface is usually smooth, and the degree of error is often small. Popular methods include the cocone method [3] and the power crust method [2] which use a piecewise linear approximation of the surface based on a Delaunay triangulation of the input points. Alternatively there is the ball-pivoting algorithm [6] which incrementally builds a surface based on the idea of a ball rolling across and intersecting points. These methods are also incredibly susceptible to noise due to their assumption that the points exactly lie on the object surface. This can be mitigated by noise smoothing or popular point sampling techniques such as Poisson disk sampling [16], but the loss of data can result in an output too abstract from the true reconstruction.

Alternative to point interpolation, there is also the category of algorithms that deal with point approximation. This removes the restriction that points must lie on the reconstruction object, and the methods are much more varied. Many of these techniques require knowledge of the point normals, which is absent information when generating input points from images. Thus methods have developed for determining and orienting the normals of points based on estimating the normal of a plane tangent to the surface [67]. Known point

normal information now introduces the use of approximation reconstruction techniques. Signed distance function approaches such as those developed by Hoppe et al. [34, 33] define a surface in implicit form and then allow for the extraction of a final triangulated mesh using popular approaches such as marching cubes [53]. Instead of using a distance function, methods such as Poisson reconstruction [40] use indicator functions to define a water tight model. Lastly there are approaches using moving least-squares (MLS) which were introduced by Levin [50] and soon popularized through point set surfaces by Alexa et al. [1] All together these form a vast library of surface reconstruction techniques to use between the interpolation and approximation subcategories.

With such a large volume of available solutions to the surface reconstruction problem, it is important to identify the best approach given a specific data set. In our case we need to identify a method that can deal with the sparsity of our point cloud volume while also not falling susceptible to noise. Other works have done similar data collection and reconstruction of underwater systems using cameras as their primary means of data collection. Johnson-Roberson et al. [38] used Delaunay triangulation to reconstruct individual top down stereo images of the sea floor. These individual reconstructions are then aggregated together using a Volumetric Range Image Processing (VRIP) technique developed by Curless and Levoy. Campos et al. [11] introduced a method for 3D surface reconstruction with an emphasis on underwater optical mapping. This method is inspired by restricted Delaunay triangulation and is robust to noise and outliers often attributed with underwater data collection. Unfortunately these two methods have clear advantages. The former only scans directly below at the more or less flat ocean floor, while the latter builds a point set from a large collection of camera angles and locations. Another study by Tischenko [79] used a range scan and collected data out of water, but compared the best approaches to reconstructing a hollow cylindrical point cloud shape. This is similar to the shape of our cave system and can thus prove useful. This study found the marching cubes implementation mentioned earlier by Hoppe to produce the most accurate reconstruction results.

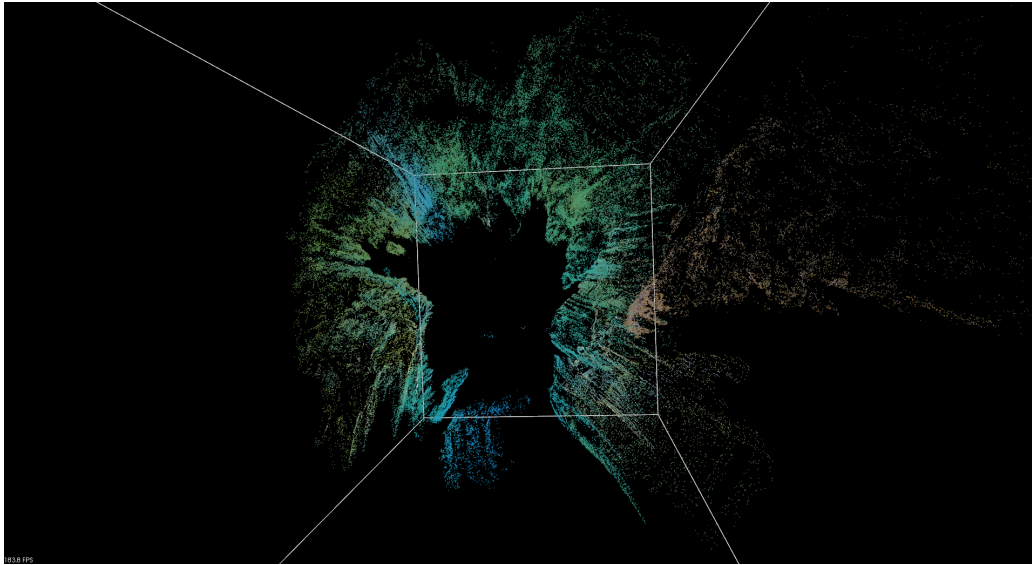
3.3 ALGORITHM COMPARISON

From a surface reconstruction perspective, the point clouds generated from our light ring model have a number of important characteristics. The only data outputted by stereo matching is the (x,y,z) coordinates of the image points in space with respect to their relative camera poses. These points can additionally coincide with RGB values from the image, but due to the nature of our data collected the colors are always some of the highest intensity pixel values and thus much of the texture detail is lost. Because no other data is currently saved with the point data, we lack information about the point normals. At the same time, there are often outlier points that make it to the stereo matching phase. One advantage of our method is that since we only project points on the ring of light which should be in a different location every frame, theoretically our points should have limited overlap. With these problem specifics and constraints defined, the task now becomes identifying a proper surface reconstruction technique to form our initial base mesh.

For the sake of rapid prototyping different reconstruction methods on our data set, tests were run using the application Meshlab. Meshlab has a number of the techniques discussed in 3.2 built in making it a quick option for comparing the outputs. Tests run in Meshlab could not be integrated into our pipeline, but they helped identify the proper procedure.

Our pipeline renders our point cloud using the Point Cloud Library(PCL). PCL has a number of additional functionalities that can help in the reconstruction phase. First off, statistical outlier removal can be applied to our points. Mean distances are computer for each point in regards to its neighboring points. Points whose mean distances exceed that of the global mean and standard deviation are labeled outliers and removed from the set. This eliminates points that were picked up by the feature detector in the image processing stage, but whos spacial position would only distort the final reconstruction. Second, we can apply the normal calculation and orientation described in 3.2. The points we projected from the image are all on the inside of the cave, so all of the normals should be oriented inwards. PCL makes the assumption that normals are oriented towards the camera, which

is exactly what we want since our camera moves through the inside of the cave. A subset of the resulting normals can be seen in 3.1.



(a)



(b)

Figure 3.1: 3.1a A point cloud segment of the ten second traversal. 3.1b The same segment with white normal lines protruding inwards

Moving this normal oriented point cloud over to Meshlab allows us to now experiment with the different reconstruction techniques. These tests were run on the points generated

from the first 100 frames of the ten second traversal video. Ultimately our reconstruction would never run on the full frame data set due to processing power constraints, so instead the cave would be generated in subsets and stitched together. 100 frames was enough to define the shape of the walls, and adding in more had minimal effect on the local outputs. Textures were generated using the RGB values of the input points. This loses a lot of the texture detail present in the images themselves, but a base model needed to be established first before creating a proper texture projection.

An initial test was done using only Delaunay triangulation. This was the procedure that had been used on ocean floor mosaicking where the overall shape of the points was flatter. Unfortunately for our point cloud, Delaunay triangulation creates a completely incomprehensible mass.

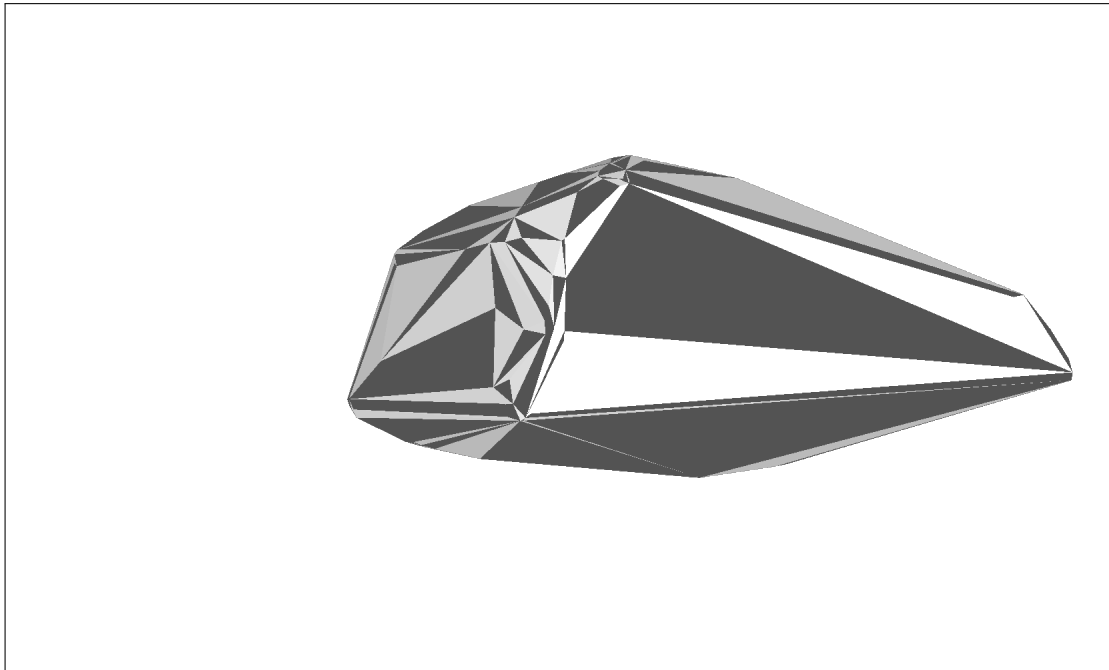


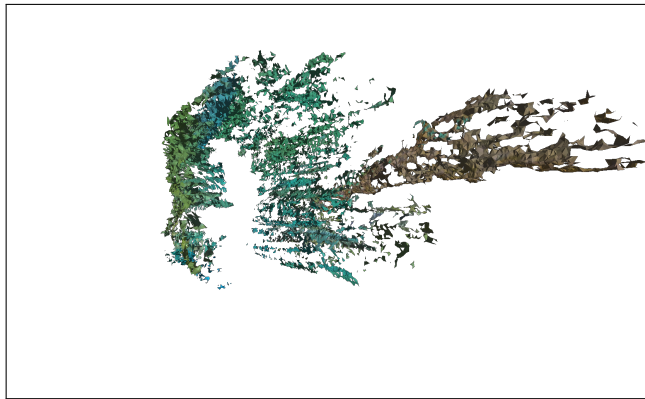
Figure 3.2: Results of running Delaunay triangulation.

If we try an interpolative method such as ball-pivoting, we see in 3.3a that the results don't look much better than Delaunay triangulation. If we take sampling into account, though, dropping the total point count down from 100,000 points to 10,000 or 1,000 produces fast low res versions of the cave mesh as seen in 3.3b and 3.3c respectively. These

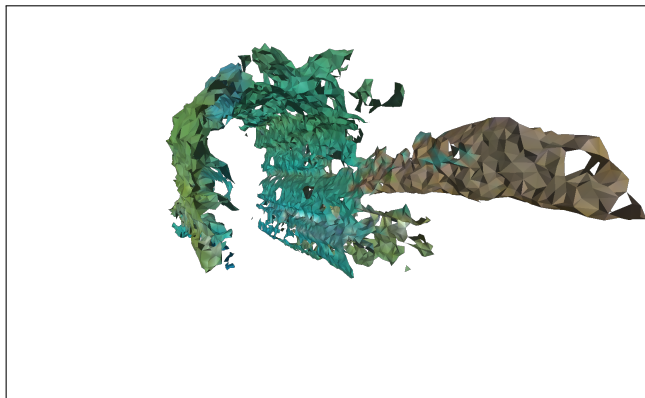
can be useful in quickly generating surfaces that have at least some resemblance of the shape.

Meshlab includes two marching-cubes based reconstruction approaches using MLS, algebraic point set surfaces (APSS) [28, 27] and robust implicit MLS (RIMLS) [61]. After adjusting the MLS filter scale, these methods generate the best reconstructions achieved. Both produce similar outputs, but the RIMLS method appears to create smoother surfaces in a more locations. These results are shown in 3.4.

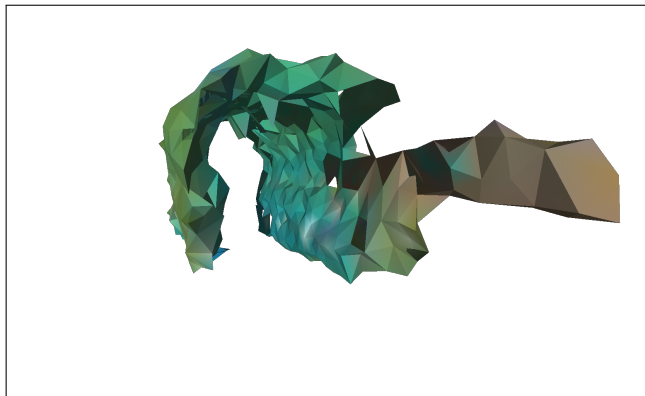
It is clear the shape of the cave is still not perfect. It is expected most of problems arise from our camera set up and lack of input data detail. Due to our small baseline, We loose a lot of accuracy as we move away from the camera. This is likely a cause for the misalignment occurring along the walls. These kinds of errors in our reprojection can cause reconstruction problems when points fall to closely to other points they shouldn't be near. Testing with other camera systems should shed light on areas of our reconstruction pipeline that can benefit from finer tuning.



(a)



(b)



(c)

Figure 3.3: 3.3a Ball-pivoting applied to the entire point set. 3.3b Applied to a 10,000 sampling. 3.3c Applied to 1,000 point sampling.

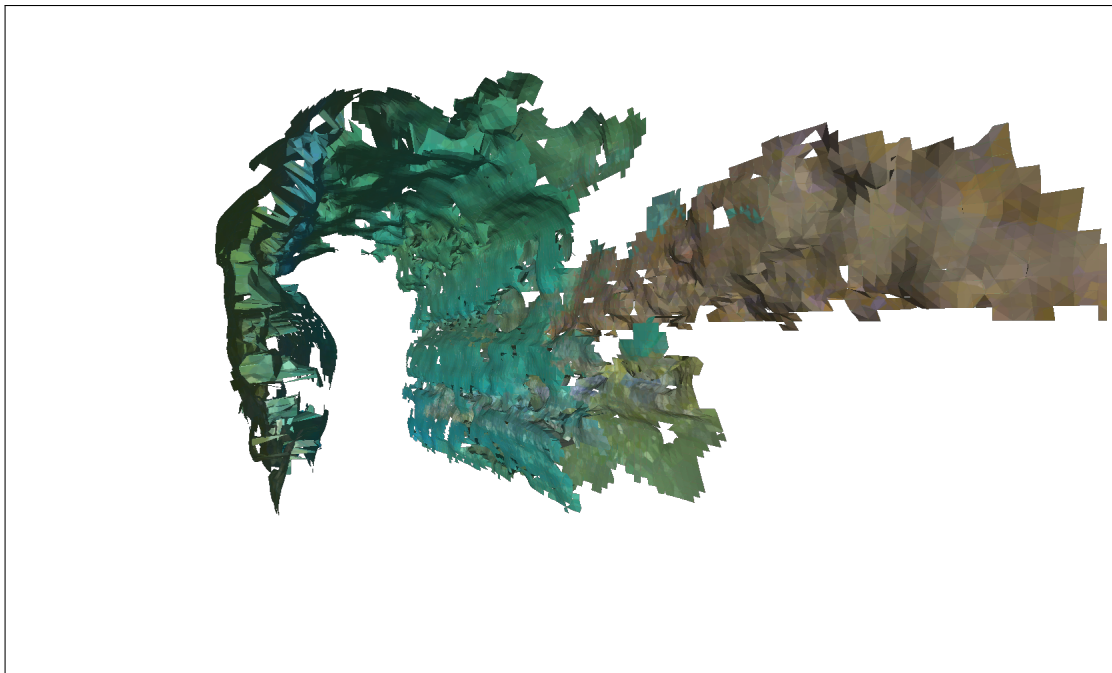


Figure 3.4: RIMLS marching cubes surface reconstruction

CONCLUSION

This thesis presented contributions to the topics of camera calibration, point cloud reconstruction, and surface reconstruction for the application of reconstructing 3-D models of underwater cave systems. In order to help calibrate camera systems with extreme distortion such as the GoPro in SuperView mode, we studied calibration techniques that took into account the removal of outlier images. MATLAB's computer vision system toolbox was found to be the most effective way to handle the problems of high distortion systems, and a simplified OpenCV version of their outlier removal was ported over. With a successfully calibrated camera we presented the first ever reconstruction results from an underwater cave using a novel approach utilizing the artificial lighting of the scene as a tool to map the boundaries. This approach was applied on actual collected cave dive footage, and was effective in both producing a reconstruction and not interfering with the standard procedure the cave divers follow. Lastly the point cloud reconstructions underwent a series of tests for surface reconstruction using a number of the most prominent techniques in the field.

We identified a number of areas for which future work can expand upon through the study of our tests and results. Different camera systems with a larger baseline and even more reduced calibration error will hopefully create more accurate reconstructions that in turn lead to more accurate surfaces. The inclusion of additional non obtrusive sensors such as sonar can also increase the accuracy of our results.

Cave mapping is a problem with significant impact across a wide range of fields. Steps made towards increasing the accuracy and efficiency of these mappings, especially at a low cost and non-cumbersome level, is vital. This work introduces a pivotal first step towards achieving complete mappings..

BIBLIOGRAPHY

- [1] Marc Alexa et al. “Computing and rendering point set surfaces”. In: *IEEE Transactions on visualization and computer graphics* 9.1 (2003), pp. 3–15.
- [2] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. “The power crust”. In: *Proceedings of the sixth ACM symposium on Solid modeling and applications*. ACM. 2001, pp. 249–266.
- [3] Nina Amenta et al. “A simple algorithm for homeomorphic surface reconstruction”. In: *Proceedings of the sixteenth annual symposium on Computational geometry*. ACM. 2000, pp. 213–222.
- [4] Caterina Balletti et al. “Calibration of action cameras for photogrammetric purposes”. In: *Sensors* 14.9 (2014), pp. 17471–17490.
- [5] H. Bay, T. Tuytelaars, and L. Van Gool. “SURF: Speeded up robust features”. In: *European Conference on Computer Vision*. 2006, pp. 404–417.
- [6] F. Bernardini et al. “The ball-pivoting algorithm for surface reconstruction”. In: *IEEE Transactions on Visualization and Computer Graphics* 5.4 (Oct. 1999), pp. 349–359. ISSN: 1077-2626. DOI: 10.1109/2945.817351.
- [7] V. Brandou et al. “3-D Reconstruction of Natural Underwater Scenes Using the Stereovision System IRIS”. In: *OCEANS 2007 - Europe*. June 2007, pp. 1–6. DOI: 10.1109/OCEANSE.2007.4302315.
- [8] D. C. Brown. “Close-range camera calibration”. In: *Photogrammetric Engineering* 37.8 (1971), pp. 855–866.
- [9] D.C. Brown. “Decentering Distortion of Lenses”. In: *Photogrammetric Engineering* 32.3 (1966), pp. 444–462.
- [10] C. McKinlay. *Woodville Karst Plain Project (WKPP)*. URL:<http://www.wkpp.org>. Apr. 2015.

- [11] Ricard Campos et al. “A surface reconstruction method for in-detail underwater 3D optical mapping”. In: *The International Journal of Robotics Research* (2014), p. 0278364914544531.
- [12] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1986), pp. 679–698.
- [13] A. E. Conrady. “Lens-systems, decentered”. In: *Monthly Notices of the Royal Astronomical Society* 79 (1919), pp. 384–390.
- [14] Peter Corke et al. “Experiments with underwater robot localization and tracking”. In: *Proc. IEEE Int. Conf. on Robotics and Automation*. 2007, pp. 4556–4561.
- [15] P. Corke et al. “Experiments with Underwater Robot Localization and Tracking”. In: *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*. 2007, pp. 4556–4561. DOI: <http://dx.doi.org/10.1109/ROBOT.2007.364181>.
- [16] Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. “Efficient and flexible sampling with blue noise properties of triangular meshes”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.6 (2012), pp. 914–924.
- [17] S. S. da Costa Botelho et al. “Visual odometry and mapping for underwater autonomous vehicles”. In: *6th Latin American Robotics Symposium (LARS)*. 2009, pp. 1–6.
- [18] M. Daum and G. Dudek. “On 3-D Surface Reconstruction Using Shape from Shadows”. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Angeles, CA, June 1998, pp. 461–468.
- [19] Jakob Engel, Thomas Schops, and Daniel Cremers. “LSD-SLAM: Large-Scale Direct Monocular SLAM”. English. In: *European Conference on Computer Vision (ECCV)*. Ed. by David Fleet et al. Vol. 8690. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 834–849. ISBN: 978-3-319-10604-5.
- [20] B. Englot and F. S. Hover. “Three-dimensional coverage planning for an underwater inspection robot”. In: *The International Journal of Robotics Research* 32.9-10 (Sept. 2013), pp. 1048–1073. ISSN: 0278-3649. DOI: [10.1177/0278364913490046](https://doi.org/10.1177/0278364913490046). URL: <http://ijr.sagepub.com/content/32/9-10/1048.refs>.
- [21] R. Eustice et al. “Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters”. In: *International Journal of Robotics Research* 25.12 (Dec. 2006), pp. 1223–1242.

- [22] D. C. Ford and P. W. Williams. *Karst Geomorphology and Hydrology*. Chapman & Hall, 1994.
- [23] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. “SVO: Fast semi-direct monocular visual odometry”. In: *IEEE Int. Conf. on Robotics and Automation*. 2014, pp. 15–22.
- [24] J.G. Fryer and D.C. Brown. “Lens Distortion for Close-Range Photogrammetry”. In: *Photogrammetric Engineering and Remote Sensing* 52.1 (1986), pp. 51–58.
- [25] Marcus Gary et al. “3D mapping and characterization of sistema Zacatón from DEPTHX (DEep Phreatic THERmal eXplorer)”. In: *Proc. of KARST08: 11th Sink-hole Conference ASCE*. 2008.
- [26] E. Gray. *Disasters of the Deep: A Comprehensive Survey of Submarine Accidents and Disasters*. Leo Cooper, 2003.
- [27] Gaël Guennebaud, Marcel Germann, and Markus Gross. “Dynamic sampling and rendering of algebraic point set surfaces”. In: *Computer Graphics Forum*. Vol. 27. 2. Wiley Online Library. 2008, pp. 653–662.
- [28] Gaël Guennebaud and Markus Gross. “Algebraic point set surfaces”. In: *ACM Transactions on Graphics (TOG)*. Vol. 26. 3. ACM. 2007, p. 23.
- [29] B. Gulden. *WORLD LONGEST UNDERWATER CAVES*. URL:<http://www.caverbob.com/uwcaves.htm>. Apr. 2015.
- [30] J. Hiekkil and O. Silven. “A Four-step Camera Calibration Procedure with Implicit Image Correction”. In: (1997).
- [31] Heiko Hirschmüller. “Stereo processing by semiglobal matching and mutual information”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 (2008), pp. 328–341.
- [32] A. Hogue, A. German, and M. Jenkin. “Underwater environment reconstruction using stereo and inertial data”. In: *IEEE International Conference on Systems, Man and Cybernetics, ISIC*. Oct. 2007, pp. 2372–2377. DOI: 10.1109/ICSMC.2007.4413666.
- [33] Hugues Hoppe. “Surface Reconstruction from Unorganized Points”. UMI Order No. GAX95-04632. PhD thesis. Seattle, WA, USA, 1995.
- [34] Hugues Hoppe et al. “Surface Reconstruction from Unorganized Points”. In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 71–78. ISBN:

0-89791-479-1. DOI: 10.1145/133994.134011. URL: <http://doi.acm.org/10.1145/133994.134011>.

- [35] F. S. Hover et al. “A Vehicle System for Autonomous Relative Survey of In-Water Ships”. In: *Marine Technology Society Journal* 41.2 (June 2007), pp. 44–55. ISSN: 00253324. DOI: 10.4031/002533207787442196.
- [36] H. Hu and G. Kantor. “Instance selection for efficient and reliable camera calibration”. In: *Proc. IEEE International Conference on Robotics and Automation*. May 2016, pp. 4335–4340. DOI: 10.1109/ICRA.2016.7487632.
- [37] Hordur Johannsson et al. “Imaging sonar-aided navigation for autonomous underwater harbor surveillance”. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. 2010, pp. 4396–4403.
- [38] Matthew Johnson-Roberson et al. “Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys”. In: *Journal of Field Robotics* 27.1 (2010), pp. 21–51.
- [39] J. Kannala and S. Brandt. “A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.8 (2006), pp. 1335–1340.
- [40] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. “Poisson Surface Reconstruction”. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP’06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, pp. 61–70. ISBN: 3-905673-36-3. URL: <http://dl.acm.org/citation.cfm?id=1281957.1281965>.
- [41] Jonathan Kelly and Gaurav S. Sukhatme. “Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-Calibration”. In: *The International Journal of Robotics Research* 30.1 (Jan. 2011), pp. 56–79. DOI: 10.1177/0278364910382802.
- [42] J. R. Kender and E. M. Smith. “Shape from darkness; deriving surface information from dynamic shadows”. In: *Proc. AAAI National Conference on Artificial Intelligence (AAAI)*. 1986, pp. 664–669.
- [43] A. Kim and R. M. Eustice. “Real-time visual SLAM for autonomous underwater hull inspection using visual saliency”. In: *IEEE Transactions on Robotics* 29.3 (2013), pp. 719–733.
- [44] G. Klein and D. Murray. “Parallel Tracking and Mapping for Small AR Workspaces”. In: *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR’07)*. Nara, Japan, Nov. 2007.

- [45] G. Klein and D. Murray. “Parallel Tracking and Mapping for Small AR Workspaces”. In: *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. Nov. 2007, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852.
- [46] K. N. Kutulakos and S. M. Seitz. “A Theory of Shape by Space Carving”. English. In: *International Journal of Computer Vision* 38.3 (2000), pp. 199–218. ISSN: 0920-5691. DOI: 10.1023/A:1008191222954. URL: <http://dx.doi.org/10.1023/A:1008191222954>.
- [47] Ed Lane. *The Spring Creek Submarine Springs Group, Wakulla County, Florida*. Tech. rep. Special Publication 47. Tallahassee, FL: Florida Geological Survey, 2001.
- [48] Chong-Moo Lee et al. “Underwater navigation system based on inertial sensor and doppler velocity log using indirect feedback Kalman filter”. In: *Int. J. of Offshore and Polar Engineering* 15.02 (2005).
- [49] A. Leone, G. Diraco, and C. Distanto. “Stereo Vision”. In: *InTech*, 2008. Chap. 11. A Stereo Vision Framework for 3-D Underwater Mosaicking, pp. 173–196. ISBN: ISBN: 978-953-7619-22-0,
- [50] David Levin. “Mesh-independent surface interpolation”. In: *Geometric modeling for scientific visualization*. Springer, 2004, pp. 37–49.
- [51] Alberto Quattrini Li et al. “Experimental Comparison of open source Vision based State Estimation Algorithms”. In: *International Symposium of Experimental Robotics (ISER)*. Tokyo, Japan, Mar. 2016.
- [52] Jorge Lobo and Jorge Dias. “Relative Pose Calibration Between Visual and Inertial Sensors”. In: *The International Journal of Robotics Research* 26.6 (2007), pp. 561–575. DOI: 10.1177/0278364907079276. eprint: <http://dx.doi.org/10.1177/0278364907079276>. URL: <http://dx.doi.org/10.1177/0278364907079276>.
- [53] William E. Lorensen and Harvey E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. In: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '87*. New York, NY, USA: ACM, 1987, pp. 163–169. ISBN: 0-89791-227-6. DOI: 10.1145/37401.37422. URL: <http://doi.acm.org/10.1145/37401.37422>.
- [54] Ian Mahon et al. “Efficient view-based SLAM using visual loop closures”. In: *IEEE Trans. on Robotics* 24.5 (2008), pp. 1002–1014.
- [55] Miquel Massot-Campos and Gabriel Oliver-Codina. “Optical Sensors and Methods for Underwater 3D Reconstruction”. In: *Sensors* 15.12 (2015), pp. 31525–31557.

- [56] F. M. Mirzaei and S. I. Roumeliotis. “A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation”. In: *IEEE Transactions on Robotics* 24.5 (Oct. 2008), pp. 1143–1156. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2004486.
- [57] R. Mur-Artal and J. D. Tardos. “ORB-SLAM: Tracking and Mapping Recognizable Features”. In: *Robotics: Science and Systems (RSS) Workshop on Multi View Geometry in Robotics (MVGRO)*. 2014.
- [58] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. “ORB-SLAM: a Versatile and Accurate Monocular SLAM System”. In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671.
- [59] L. Nalpantidis and A. Gasteratos. “Stereo vision for robotic applications in the presence of non-ideal lighting conditions”. In: *Image and Vision Computing* 28.6 (2010), pp. 940–951. ISSN: 0262-8856. DOI: <http://dx.doi.org/10.1016/j.imavis.2009.11.011>. URL: <http://www.sciencedirect.com/science/article/pii/S0262885609002674>.
- [60] D. Nistér, O. Naroditsky, and J. Bergen. “Visual odometry”. In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2004, pp. I–652.
- [61] A Cengiz Öztireli, Gael Guennebaud, and Markus Gross. “Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression”. In: *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library. 2009, pp. 493–501.
- [62] D. Ribas, P. Ridao, and J. Domingo. “Underwater SLAM in Man-Made Structured”. In: *Journal of Field Robotics* 25 (2008), pp. 898–921. DOI: 10.1002/rob.
- [63] Andrew Richardson, Johannes Strom, and Edwin Olson. “AprilCal: Assisted and repeatable camera calibration”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013.
- [64] P. Ridao et al. “Visual inspection of hydroelectric dams using an autonomous underwater vehicle”. In: *Journal of Field Robotics* 27.6 (2010), p. 759 778.
- [65] Paul Rigby, Oscar Pizarro, and Stefan B Williams. “Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements”. In: *MTS/IEEE OCEANS*. 2006, pp. 1–6.
- [66] M. Rufli, D. Scaramuzza, and R. Siegwart. “Automatic Detection of Checkerboards on Blurred and Distorted Images”. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2008), pp. 3121–3126.

- [67] Radu Bogdan Rusu. “Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments”. PhD thesis. Computer Science department, Technische Universitaet Muenchen, Germany, Oct. 2009.
- [68] D. Scaramuzza. “Omnidirectional Vision: from Calibration to Robot Motion Estimation”. PhD thesis. ETH Zurich, 2008.
- [69] D. Scaramuzza, A. Martinelli, and R. Siegwart. “A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion”. In: *IEEE International Conference of Vision Systems* (2006), pp. 45–52.
- [70] D. Scaramuzza, A. Martinelli, and R. Siegwart. “A Toolbox for Easy Calibrating Omnidirectional Cameras”. In: *IEEE International Conference on Intelligent Robots and Systems (IROS)* (2006), pp. 5695–5701.
- [71] Val E Schmidt and Yuri Rzhanov. “Measurement of micro-bathymetry with a GO-PRO underwater stereo camera pair”. In: *Oceans, 2012*. IEEE. 2012, pp. 1–6.
- [72] Johannes Lutz Schönberger and Jan-Michael Frahm. “Structure-from-Motion Revisited”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [73] A. Sedlazeck and R. Koch. “Perspective and Non-perspective Camera Models in Underwater Imaging - Overview and Error Analysis”. English. In: *Outdoor and Large-Scale Real-World Scene Analysis*. Ed. by F. Dellaert et al. Vol. 7474. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 212–242. ISBN: 978-3-642-34090-1. DOI: 10.1007/978-3-642-34091-8_10. URL: http://dx.doi.org/10.1007/978-3-642-34091-8_10.
- [74] J. Servos, M. Smart, and S. L. Waslander. “Underwater stereo SLAM with refraction correction”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nov. 2013, pp. 3350–3355. DOI: 10.1109/IROS.2013.6696833.
- [75] J. Shi and C. Tomasi. “Good Features to Track”. In: *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’94)*. 1994, pp. 593–600.
- [76] F. Shkurti, I. Rekleitis, and G. Dudek. “Feature Tracking Evaluation for Pose Estimation in Underwater Environments”. In: *Canadian Conference on Computer and Robot Vision*. St. J., NF Canada, 2011, pp. 160–167.
- [77] Mark Shortis. “Calibration techniques for accurate measurements by underwater camera systems”. In: *Sensors* 15.12 (2015), pp. 30810–30826.

- [78] Jeff Snyder. “Doppler Velocity Log (DVL) navigation for observation-class ROVs”. In: *MTS/IEEE SEATTLE OCEANS*. 2010, pp. 1–9.
- [79] Inna Tishchenko. “Surface Reconstruction from Point Clouds”. Bachelor Thesis. Swiss Federal Institute of Technology, Zurich, 2010.
- [80] R. Y. Tsai. “A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses”. In: *IEEE J. Robotics Automat.* 3.4 (1987), pp. 323–334.
- [81] C. White et al. “The Malta cistern mapping project: Underwater robot mapping and localization within ancient tunnel systems”. In: *Journal of Field Robotics* 27.4 (2010), p. 399 411.
- [82] Marios Xanthidis, Alberto Quattrini Li, and Ioannis Rekleitis. “Shallow Coral Reef Surveying by Inexpensive Drifters”. In: *MTS/IEEE Oceans*. Shanghai, China, Apr. 2016.
- [83] Zexuan Xu et al. “Long distance seawater intrusion through a karst conduit network in the Woodville Karst Plain, Florida”. In: *Scientific Reports* 6 (Aug. 2016), pp. 1–10.
- [84] Z. Zhang. “A flexible new technique for camera calibration ”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.11 (2000), pp. 1330–1334.