A Multi-Sensor fusion based Underwater SLAM System

by

Sharmin Rahman

Bachelor of Science
Military Institute of Science and Technology 2012

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Computer Science and Engineering

College of Engineering and Computing

University of South Carolina

2019

Accepted by:

Ioannis Rekleitis, Major Professor

Jason O'Kane, Committee Member

Song Wang, Committee Member

John R. Rose, Committee Member

Nikolaos Vitzilaios, Committee Member

Dr., Cheryl Addy

© Copyright by Sharmin Rahman, 2019
All Rights Reserved.

Dedication

Acknowledgments

Abstract

Exploration of underwater environments with autonomous robots could assist us in a variety of scenarios, ranging from historical studies to health monitoring of coral reef; underwater infrastructure inspection e.g., bridges, hydroelectric dams, water supply systems and oil rigs. Mapping underwater structures is important in several fields, such as, marine archaeology, Search and Rescue (SaR), resource management, hydrogeology, and speleology. However, due to the highly unstructured nature of such environments, navigation by human divers could be extremely dangerous, tedious and labor intensive. Hence, employing an underwater robot is an excellent fit to build the map of the environment while simultaneously localizing itself in the map.

The contribution of this thesis is the design and development of a real-time robust Simultaneous Localization and Mapping (SLAM) algorithm for underwater domain. A novel tightly-coupled keyframe-based non-linear optimization framework with loop-closing and relocalization capabilities fusing Sonar, Visual, Inertial and Depth information has been presented. Introducing acoustic range information to aid the visual data in underwater, shows improved reconstruction. The availability of depth information from water pressure enables a robust initialization and refines the scale; as well as assists to reduce the drift due to the tightly-coupled formulation. In addition, we propose to augment the pipeline with magnetometer for a more accurate orientation estimation from the dead reckoning sensor. To address the denser reconstruction of the surroundings in a low lighting conditions, a contour-based reconstruction approach utilizing the well defined edges between the well lit areas and darkness has been developed. Furthermore, we propose a semi-direct sparse approach of recon-

struction by jointly minimizing the photometric and reprojection error from direct method and indirect method respectively where indirect method is used for accurate tracking while high-gradient pixels help in reconstruction. Experimental results on datasets collected with a custom-made underwater sensor suite and an autonomous underwater vehicle (AUV) Aqua2 from challenging underwater environments with poor visibility demonstrate performance never achieved before in terms of accuracy and robustness.

Preface

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Exploring and mapping underwater environments such as caves, bridges, dams, and shipwrecks, are extremely important tasks for the economy, conservation, and scientific discoveries. Currently, most of the efforts are performed by divers that need to take measurements manually using a grid and measuring tape, or using hand-held sensors Henderson et al. 2013, and data is post-processed afterwards. Autonomous Underwater Vehicles (AUVs) present unique opportunities to automate this process; however, there are several open problems that still need to be addressed for reliable deployments, including real-time robust Simultaneous Localization and Mapping (SLAM), the focus of this thesis.

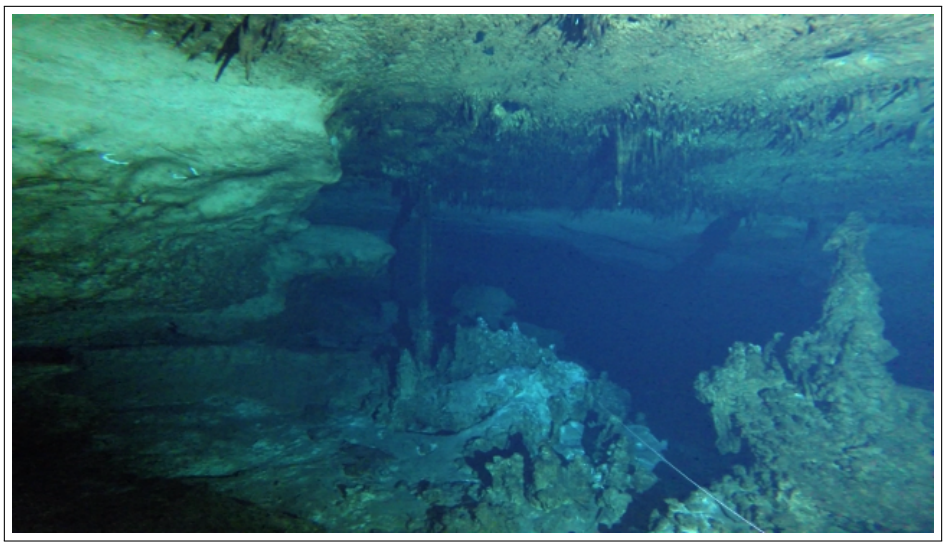


Figure 1.1: Typical scene from an underwater cave.

1.1 Motivation

The underwater environment presents unique challenges to vision-based state estimation. In particular, suspended particulates, blurriness, and light and color attenuation result in features that are not as clearly defined as above water. Consequently results from different vision-based state estimation packages show a significant number of outliers resulting in inaccurate estimate or even complete tracking loss. Here we present a comprehensive study of the performances of state-of-the-art open-source Visual and Visual-Inertial state estimation algorithms in underwater domain and draw out the scope of improvements by introducing acoustic and pressure sensor.

1.1.1 Performances of state-of-the-art Visual and Visual-Inertial State Estimation Algorithms in Underwater

We have considered ten state estimation packages which are characterised by the following:

- *number of cameras*, e.g., monocular, stereo, or more rarely multiple cameras;
- *the presence of an IMU*;
- *direct* vs. *indirect* (feature-based) methods;
- *loosely* vs. *tightly-coupled* optimization when multiple sensors are used – e.g., camera and IMU;
- the presence of a *loop closing* mechanism.

Table 1.1 lists the methods evaluated and their properties.

Datasets

Most of the standard benchmark datasets represent only a single scenario, such as a lab space (e.g. [Sturm et al. 2012; Burri et al. 2016]), or a urban environment (e.g.

Kitti in [Geiger et al. 2013]), and with good visual quality. The limited nature of the public datasets is one of the primary motivations to evaluate these packages with datasets collected by our lab over the years in more challenging environments, such as underwater.

In particular, the datasets used can be categorized according to the robotic platform used:

- Underwater sensor suite operated by a diver around a sunken bus (Fantasy Lake, North Carolina) – see Fig. 1.2(a),(b) – and inside an underwater cave (Ginnie Springs, Florida); see Fig. 1.2(c). The custom-made underwater sensor suite is equipped with an IMU operating at 100 Hz (MicroStrain 3DM-GX15) and a stereo camera running at 15 fps, 1600×1200 (IDS UI-3251LE).

- Underwater sensor suite mounted on an Diver Propulsion Vehicle (DPV). Data collected over the coral reefs of Barbados; see Fig. 1.2(d).

- Aqua2 Autonomous Underwater Vehicle (AUV) over a coral reef (Fig. 1.2(e)) and an underwater structure (Lake Jocassee, South Carolina) (Fig. 1.2(f)), with the same setup as the underwater sensor suite.

The overall performance of the tested packages is discussed next. LSD-SLAM in [Engel, Schöps, and Cremers 2014], REBiVO in [Tarrio and Pedre 2017], and

Table 1.1: Summary of characteristics for evaluated methods.

Method		**Camera**	**IMU**	**Indirect/ Direct**	**(L)oosely/ (T)ightly**	**Loop Closure**
LSD-SLAM	Engel, Schöps, and Cremers 2014	mono	no	direct	N/A	yes
DSO	Engel, Koltun, and Cremers 2018	mono	no	direct	N/A	no
SVO	Forster et al. 2017b	multi	optional	semi-direct	N/A	no
ORB-SLAM2	Mur-Artal, Montiel, and Juan D. Tardós 2015a	mono, stereo	no	indirect	N/A	yes
REBiVO	Tarrio and Pedre 2017	mono	optional	indirect	L	no
Mono-MSCKF	Research group of Prof. Kostas Daniilidis 2018	mono	yes	indirect	T	no
Stereo-MSCKF	Sun et al. 2018	stereo	yes	indirect	T	no
ROVIO	Bloesch et al. 2017	multi	yes	direct	T	no
OKVIS	Leutenegger et al. 2015	multi	yes	indirect	T	no
VINS-Mono	Qin, Li, and Shen 2018	mono	yes	indirect	T	yes

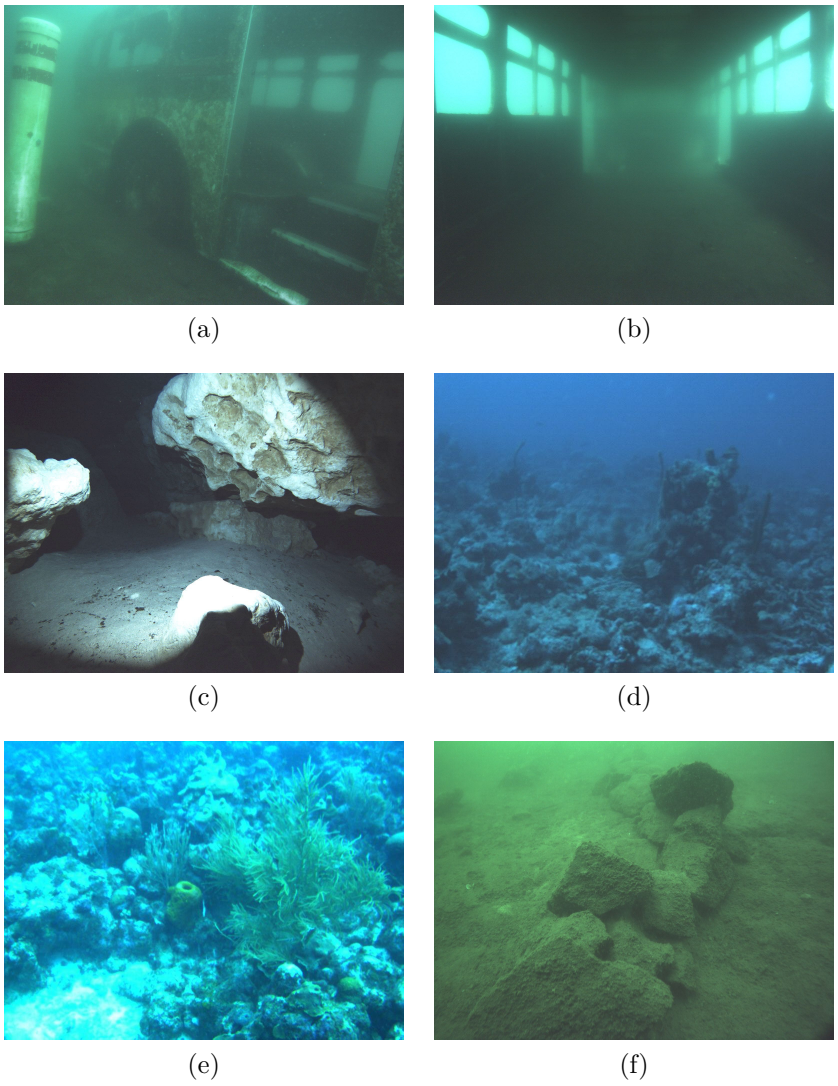


Figure 1.2: Sample images from the evaluated datasets. (a) UW sensor suite outside a sunken bus (NC); (b) UW sensor suite inside a sunken bus (NC); (c) UW sensor suite inside a cave (FL); (d) UW sensor suite mounted on a Diver Propulsion Vehicle (DPV) over a coral reef; (e) Aqua2 AUV over a coral reef; (f) AUV over a fake cemetery (SC).

Monocular SVO were unable to produce any consistent results, as such, they were excluded from Table 1.2.

DSO in [Engel, Koltun, and Cremers 2018] requires full photometric calibration accounting for the exposure time, lens vignetting and non-linear gamma response function for best performance. Even without photometric calibration, it worked well

	Monocular			Monocular + IMU						Stereo			Stereo + IMU		
	dso	orbslam	orbslamlc	msckf	okvis	rovio	svo	vinsmono	vinsmonolc	orbslam	orbslamlc	svo	okvis	svo	stereomsckf
Bus/Out															
Cave															
Aqua2Lake															
Bus/In															
DPV															
Aqua2Reef															

Table 1.2: Performance of the different open source packages. Datasets: UW sensor suite outside a sunken bus (Bus/Out); UW sensor suite inside a cave (Cave); Aqua2 (AUV) over a fake cemetery (Aqua2Lake) at Lake Jocassee; UW sensor suite inside a sunken bus (Bus/In); UW sensor suite mounted on a Diver Propulsion Vehicle over a coral reef (DPV); Aqua2 AUV over a coral reef (Aqua2Reef). Qualitative analysis: the color chart legend is: red–failure; orange–partial failure; yellow–partial success; green-success.

on areas having high intensity gradients and when subjected to large rotation. In addition, it provided excellent reconstructions; however, in the areas with low gradient images, it was able to spatially track the motion only for a few seconds. Some scale change was also observed due to being monocular. DSO requires more computational power and memory usage compared to the other packages, which is justifiable since it uses direct method for visual odometry.

SVO 2.0 in [Forster et al. 2017b] was able to track the camera pose over long trajectories, even in parts with few features. It tracks features using the direct method by creating a depth scene. In case of low gradient images, it was subject to depth scale changes, which was predominant in mono camera where tracking failed. SVO in stereo mode without inertial measurements was able to track most of the time but was subject to rotation errors. SVO stereo with IMU was able to keep track most of the time generating a good trajectory estimate.

ORB-SLAM2 in [Mur-Artal, Montiel, and Juan D. Tardós 2015a] mono could not

initialize in both datasets collected of the sunken bus (Bus/In, Bus/Out). ORB-SLAM2 works fine in the other datasets, but loses track in some cases when running it without loop closure. With loop closure, even if the track is lost, loops can be detected and the robot can relocalize. This makes ORB-SLAM2 more robust to track loss.

Mono-MSCKF in [Research group of Prof. Kostas Daniilidis 2018] performed well when the AUV or sensor suite were standing still so that the IMU could properly initialize, otherwise it did not track. Moreover, it was among the most efficient in terms of CPU and memory usage.

ROVIO in [Bloesch et al. 2017] is one of the most efficient packages tested. Its overall performance was robust on most datasets even when just a few good features were tracked. On the Aqua2Reef dataset though, not enough features were visible and thus it could not track the trajectory.

OKVIS in [Leutenegger et al. 2015] provided good results for both monocular and stereo. In Bus/Out, despite the haze and low-contrast, OKVIS was able to detect good features and track them successfully. Also in Cave, it kept track successfully and produced accurate trajectory even in the presence of low illumination.

VINS-Mono in [Qin, Li, and Shen 2018] works well in good illumination where there are good features to track. It was one of the few packages that worked successfully in the underwater domain. In the case of Aqua2Reef, it cannot detect and track enough features and diverges. With the loop-closure module enabled, VINS-Mono reduces the drift accumulated over time in the *pose* estimate and produces a globally consistent trajectory.

Stereo-MSCKF in [Sun et al. 2018] uses the Observability Constrained EKF (OC-EKF) in [Hesch et al. 2012], which does not heavily depend on an accurate initial estimation. Also, the camera poses in the state vector can be represented with respect to the inertial frame instead of the latest IMU frame so that the uncertainty of

the existing camera states in the state vector is not affected by the uncertainty of the latest IMU state during the propagation step. As a result, Stereo-MSCKF can initialize well enough even without a perfect stand still period. It uses the first 200 IMU measurements for initialization and is recommended to not have fast motion during this period. Stereo-MSCKF worked acceptably well in most datasets except Aqua2Reef and DPV. The Stereo-MSCKF cannot initialize well over the coral reef due to the fast motion from the start and the low number of feature points. On the DPV dataset, it diverges after one-fourth time is completed.

1.1.2 Discussion

Underwater state estimation has many open challenges, including visibility, color attenuation Skaff, Clark, and Rekleitis 2008, floating particulates, blurriness, varying illumination, and lack of features Oliver, Hou, and Wang 2010. Indeed, in some underwater environments, there is a very low visibility that prevents seeing objects that are only a few meters away. This can be observed for example in Bus/Out, where the back of the submerged bus is not clearly visible. Such challenges make underwater localization very challenging, leaving an interesting gap to be investigated in the current state of the art. In addition, light attenuates with depth, with different wavelengths of the ambient light being absorbed very quickly – e.g., the red wavelength is almost completely absorbed at 5 m. This results in a change in appearance of the image, which will affect feature tracking, even in grayscale.

The appearance of color underwater is different than above, including the color loss with depth. There is a concern when most color shifts to blue, there is a loss of sharpness, which further degrades performance. This will be a venue for further research in the future, in order to investigate the effect of any color restoration to the state estimation process.

From the experimental results it was clear that direct VO approaches are not

robust as there are often no discernible features. As such DSO and SVO, quite often fail to track the complete trajectory, however, they had the best reconstructions for the tracked parts. Similar approaches that depend on the existence of a specific feature, such as edges, are not appropriate in underwater environments in general. Overall, as expected, stereo performed better than monocular, the introduction to loop closure enabled the VO/VIO packages to track for longer periods of time, and the introduction of inertial data improved the scale estimations.

1.2 Challenges in Underwater

Underwater environments suffer from poor visibility, light and color attenuation. In addition to that, floating particulates, haze, and varying illumination make vision-based state estimation almost impossible to work.

Navigation and mapping around underwater structures is very challenging; target domains include wrecks (ships, planes, and buses); underwater structures such as bridges and dams; and underwater caves. One of the primary motivations of this work is the mapping of underwater caves where exploration by human divers is an extremely dangerous operation due to the harsh environment. Figure 1.1 shows a typical cave segment. In addition to underwater vision constraints–e.g., light and color attenuation–cave environments suffer from the absence of natural illumination. Currently, for surveying of newly explored area, divers manually measure 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. This 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. Therefore, employing robotic technology to map the cave would reduce the cognitive load of divers.

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 [Ford and P. W. Williams 1994] 25% of the world's population relies on karst water resources. Our work is motivated by the Woodville Karst Plain (WKP) which is a geomorphic region that extends from Central Leon County around the "Big Bend" of Florida [Lane 2001]. 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 [C. McKinlay 2015], proving the cave system to be the longest in USA [Gulden 2015]. 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. 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. [Xu et al. 2016]. 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.

1.3 Contributions

The focus of this thesis is the robust tightly-coupled formulation of an underwater SLAM system combining acoustic data from Sonar; stereo vision; angular velocity and linear acceleration from IMU; and depth data from water pressure measurement.

A robust SLAM system combining Sonar, Visual, Inertial and Depth information. We propose a tightly-coupled keyframe based SLAM system fusing Sonar, Visual, Inertial and Depth information in a non-linear optimization-based framework for underwater domain. The underwater domain presents unique challenges in the quality of the visual data available; as such, augmenting the exteroceptive sensing with acoustic range data results in improved reconstructions of the underwater structures. Depth data from water pressure measurement enables to bound the localization error. To address drift and loss of localization – one of the main problems affecting other packages in underwater domain – a robust initialization method to refine scale using depth measurements, a fast preprocessing step to enhance the image quality, and a real-time loop-closing and relocalization method using bag of words (BoW) have been provided. Lastly we propose to augment our SVIn2 pipeline with magnetometer which would provide accurate *heading* information and thus assist in achieving robust dead-reckoning pose estimation from IMU. An *ablation study* to understand the contribution of each sensor in state estimation will also be reported. To validate the robustness and accuracy of our approach, we deployed an autonomous underwater vehicle (AUV) Aqua2 running our method on-board. Further datasets were collected with a custom-made underwater sensor suite both on hand-held mode while diving and deploying with a Diver Propulsion Vehicle (DPV). Experimental results from underwater wrecks, an underwater cave, fake underwater cemetery, over coral reef, and a submerged bus demonstrate the performance of our approach.

A contour-based reconstruction of underwater environment. Another contribution is contour-based real-time reconstruction of an underwater environment

using Sonar, Visual, Inertial, and Depth data. In particular, low lighting conditions, or even complete absence of natural light inside caves, results in strong lighting variations, e.g., the cone of the artificial video light intersecting underwater structures, and the shadow contours. The proposed method utilizes the well defined edges between well lit areas and darkness to provide additional features, resulting into a denser 3D point cloud than the usual point clouds from a Visual SLAM system. Experimental results in an underwater cave at Ginnie Springs, FL, with a custom-made underwater sensor suite demonstrate the performance of our system. This will enable more robust navigation of AUVs using the denser 3D point cloud to detect obstacles and achieve higher resolution reconstructions.

A semi-direct sparse reconstruction. Lastly, to take another step ahead from contour based reconstruction, we propose a semi-dense reconstruction to achieve a denser map of the environment along with robust odometry in real-time. Direct methods provide promising reconstruction, but due to the brightness consistency assumption, they often fail to track in challenging low-contrast environment. Hence, combining direct method and feature-based method could benefit each other – i.e., achieving a denser reconstruction from the gradient-rich pixels on the contour using direct method and tracking based on the feature based method – by jointly minimizing the photometric error and reprojection error.

Chapter 2

Related Work

2.1 Acoustic Sensor based Underwater Navigation

Sonar based underwater SLAM and navigation systems have been exploited for many years. Folkesson et al. in [Folkesson et al. 2007] used a blazed array sonar for real-time feature tracking. A feature reacquisition system with a low-cost sonar and navigation sensors was described in in [Fallon et al. 2013]. More recently, Sunfish in [Richmond et al. 2018] – an underwater SLAM system using a multibeam sonar, an underwater dead-reckoning system based on a fiber-optic gyroscope (FOG) IMU, acoustic DVL, and pressure-depth sensors – has been developed for autonomous cave exploration. Vision and visual-inertial based SLAM systems also developed in in [Salvi et al. 2008; Beall et al. 2011; Shkurti et al. 2011] for underwater reconstruction and navigation.

Most of the underwater navigation algorithms in [Leonard and Durrant-Whyte 2012], in [Lee et al. 2005], in [Snyder 2010], in [Johannsson et al. 2010], in [Rigby, Pizarro, and S. B. Williams 2006 are based on acoustic sensors such as DVL, USBL, and sonar. Nevertheless, collecting data using DVL, sonar, and USBL while diving is expensive and sometimes not suitable in challenging underwater environment, e.g., cave. Corke et al. in [Corke et al. 2007] compared acoustic and visual methods for underwater localization showing the viability of using visual methods underwater in some scenarios.

2.1.1 Underwater Cave Exploration

Visual odometry in underwater cave environment is a challenging problem due to the lack of natural light illumination and dynamic obstacles in addition to the underwater vision constraints i.e. light and color attenuation. There are not many works for mapping and localization in an underwater cave.

Robotic exploration of underwater caves is at its infancy. One of the first attempts was to explore a Cenote, a vertical shaft filled with water in [Gary et al. 2008], by the vehicle DEPTHX (DEep Phreatic THermal eXplorer) in [Stone 2007] designed

by Stone Aerospace in [Stone Aerospace 2015], equipped with LIDAR and sonar. More recently, Mallios demonstrated the first results of an Autonomous Underwater Vehicle (AUV) performing limited penetration, inside a cave [Mallios et al. 2016]. The main sensor used for SLAM is a horizontally mounted scanning sonar. A robotic fish was proposed for discovering underwater cave entrances based on vision and perform visual servoing, with experiments restricted to a swimming pool in [Chen and Yu 2014]. More recently, Sunfish in [Richmond et al. 2018] – an underwater SLAM system using a multibeam sonar, an underwater dead-reckoning system based on a fiber-optic gyroscope (FOG) IMU, acoustic DVL, and pressure-depth sensors – has been developed for autonomous cave exploration. The design of the sensor suite we use is driven by portability requirements that divers have in [Rahman, Quattrini Li, and Rekleitis 2018a], not permitting the use of some sensors, such as multibeam sonar or DVL.

2.2 Pure Visual Odometry (VO)

The literature presents many vision-based state estimation techniques, which use either *monocular* or *stereo* cameras and that are *indirect* (feature-based), *direct*, or *semi-direct* methods. For example, MonoSLAM in [Davison et al. 2007], PTAM in [Klein and Murray 2007], and ORB-SLAM in [Mur-Artal, Montiel, and Juan D. Tardós 2015b] are feature-based, LSD-SLAM in [Engel, Schöps, and Cremers 2014], and DSO in [Engel, Koltun, and Cremers 2018] are direct, and SVO in [Forster et al. 2017b] is semi-direct.

2.2.1 Direct Method

Direct methods compare the intensity values in the image and optimize the *photometric error*

Recently, direct methods (e.g., LSD-SLAM in [Engel, Schöps, and Cremers 2014],

DSO in [Engel, Koltun, and Cremers 2018]) and semi-direct method (SVO in [Forster et al. 2017b]) based SLAM systems show promising performance in 3-D reconstruction of large-scale map in real time, as well as accurate pose estimation based on direct image alignment. However, theses methods are sensitive to *brightness consistency* assumption which limits the baseline of the matches and in low visibility with small contrast environment like underwater, often result into tracking loss in [Joshi et al. 2019]. In addition, direct method suffers in presence of strong geometric noise, such as rolling shutter. For good reconstruction, they require perfect photometric calibration for modeling gain and exposure. DSO in [Engel, Koltun, and Cremers 2018] shows an improvement in performance providing a full photometric calibration that accounts for lens attenuation, gamma correction, and known exposure times. In purely monocular vision based direct SLAM, like DSO, the initialization is slow and requires very small rotational change.

2.2.2 Semi-direct Method

Semi-Direct Visual Odometry, e.g., SVO [Forster et al. 2017b] relies on direct method for tracking and triangulating pixels with high image gradients and a feature-based method for jointly optimizing structure and motion. It uses the IMU prior for image alignment and can be generalized to multi-camera systems.

2.2.3 Indirect Method

Feature-based methods pre-process images to find corners and establish correspondences, then optimize the *geometric error*.

PTAM [Klein and Murray 2007] is one of the early SLAM approaches which proposed to split the tracking and mapping process for a small AR workspace without any prior knowledge of the scene. MonoSLAM [Davison et al. 2007] is a monocular vision based real-time SLAM approach which includes an *active* approach to

mapping and measurement, a general motion model for smooth camera movement, and solutions for monocular feature initialization and feature orientation estimation. Currently ORB-SLAM [Mur-Artal, Montiel, and Juan D. Tardós 2015b] is one of the most reliable vision-based SLAM systems with loop-closing and relocalization capabilities.

2.3 Vision combined with other sensors

2.3.1 Visual-Inertial Odometry (VIO)

In the following, we highlight some of the state estimation systems which use visual-inertial measurements and feature-based method.

To improve the pose estimate, vision-based state estimation techniques have been augmented with IMU sensors, whose data is fused together with visual information. A class of approaches is based on the *Kalman Filter*, e.g., Multi-State Constraint Kalman Filter (MSCKF) in [Mourikis and Roumeliotis 2007] and its stereo extension in [Sun et al. 2018]; ROVIO in [Bloesch et al. 2017]; REBiVO in [Tarrio and Pedre 2017]. The other spectrum of methods optimizes the sensor states, possibly within a window, formulating the problem as a *graph optimization* problem. For feature-based visual-inertial systems, as in OKVIS in [Leutenegger et al. 2015] and Visual-Inertial ORB-SLAM in [Mur-Artal and Juan D Tardós 2017], the optimization function includes the IMU error term and the reprojection error. The *frontend* tracking mechanism maintains a local map of features in a marginalization window which are never used again once out of the window. VINS-Mono in [Qin, Li, and Shen 2018] uses a similar approach and maintains a minimum number of features for each image and existing features are tracked by Kanade-Lucas-Tomasi (KLT) sparse optical flow algorithm in local window. Delmerico and Scaramuzza in [Delmerico and Scaramuzza 2018] did a comprehensive comparison specifically monitoring resource usage by the different methods. While KLT sparse features allows VINS-Mono run-

ning in real-time on low-cost embedded systems, often results into tracking failure in challenging environments, e.g., underwater environments with low visibility. In addition, for loop detection additional features and their descriptors are needed to be computed for keyframes.

To avoid scale ambiguity in monocular system, stereo camera pairs are used. Oskiper et al. in [Oskiper et al. 2007] proposed a real-time VO using two pairs of backward and forward looking stereo cameras and an IMU in GPS denied environments. Howard in [Howard 2008] presented a real-time stereo VO for autonomous ground vehicles. This approach is based on *inlier detection*— i.e., using a rigidity constraint on the 3D location of features before computing the motion estimate between frames. Konolige et al. in [Konolige, Agrawal, and Sola 2010] presented a real-time large scale VO on rough outdoor terrain integrating stereo images with IMU measurements. Kitt et al. in [Kitt, Geiger, and Lategahn 2010] presented a visual odometry based only on stereo images using the trifocal geometry between image triples and a RANSAC based outlier rejection scheme. Their method requires only a known camera geometry where no rectification is needed for the images. Badino et al. in [Badino, Yamamoto, and Kanade 2013] proposed a new technique for improved motion estimation by using the whole history of tracked features for real-time stereo VO.

2.4 Visual or Visual-Inertial SLAM with Loop-Closing

Loop closure – the capability of recognizing a place that was seen before – is an important component to mitigate the drift of the state estimate. FAB-MAP in [Cummins and Newman 2008; Cummins and Newman 2011] is an appearance-based method to recognize places in a probabilistic framework. ORB-SLAM in [Mur-Artal, Montiel, and Juan D. Tardós 2015b] and its extension with IMU in [Mur-Artal and Juan D Tardós 2017] use bag-of-words (BoW) for loop closure and relocalization. VINS-Mono

also uses a BoW approach.

Note that all visual-inertial state estimation systems require a proper *initialization*. VINS-Mono uses a loosely-coupled sensor fusion method to align monocular vision with inertial measurement for estimator initialization. ORB-SLAM with IMU in [Mur-Artal and Juan D Tardós 2017] performs initialization by first running a monocular SLAM to observe the pose first and then, IMU biases are also estimated.

Given the modularity of OKVIS for adding new sensors and robustness in tracking in underwater environment – we fused sonar data in previous work in [Rahman, Quattrini Li, and Rekleitis 2018b] – we extend OKVIS to include also depth estimate, loop closure capabilities, and a more robust initialization to specifically target underwater environments.

2.5 Structure-from-Motion (SfM)

Structure-from-Motion (SfM) from unstructured collections of photographs to build the 3-D model of the scene has been addressed in different solutions, *Bundler* in [Snavely, Seitz, and Szeliski 2006] and *VisualSFM* in [C. Wu 2013]. They provided algorithmic analysis to improve computational complexity and performance accuracy. *COLMAP* in [Schonberger and Frahm 2016] proposes a SfM algorithm to improve on the state-of-the-art incremental SfM methods for 3D reconstruction from unordered image collections. They provide scene graph augmentation, a next best view selection mechanism, and an efficient triangulation and Bundle Adjustment (BA) technique. COLMAP outperforms state-of-the-art SfM system on benchmark datasets with a large number of photos from Internet with varying camera density and distributed over large area.

2.6 Multiview Stereo (MVS)

Multiview Stereo (MVS) is another well known method for reconstruction. Merrell in [Merrell et al. 2007] presented a *viewpoint-based* approach to fuse multiple stereo depth maps for reconstructing 3-D shape from video. By decoupling the processing into two stages, they are able to run large-scale reconstruction in real-time using a GPU implementation for efficient computation. The computational power available on board of the robot is very limited, making the deployment of bundle adjustment based methods not feasible on the robot.

2.7 Vision-based Underwater Navigation

Exploiting SLAM techniques in underwater environment is a difficult task due to the highly unstructured nature. Salvi et al. in [Salvi et al. 2008] implemented a real-time EKF-SLAM incorporating a sparsely distributed robust feature selection and 6-DOF pose estimation using only calibrated stereo cameras. Johnson et al. in [Johnson-Roberson et al. 2010] proposed an idea to generate 3D model of the seafloor from stereo images. Beall et al. in [Beall et al. 2011] presented an accurate 3D reconstruction on a large-scale underwater dataset by performing bundle adjustment over all cameras and a subset of features rather than using a traditional filtering technique. A stereo SLAM framework named *selective* SLAM (SSLAM) for autonomous underwater vehicle localization was proposed in in [Bellavia, Fanfani, and Colombo 2015].

Vision is often combined with IMU and other sensors in underwater domain for improved estimation of pose. Hogue et al. in [Hogue, German, and Jenkin 2007] used stereo and IMU for underwater reconstruction. Stereo and IMU were used for VO in in [Hildebrandt and Kirchner 2010] and in [Wirth, Carrasco, and Codina 2013]. Sáez et al. in [Sáez et al. 2006] proposed a 6DOF Entropy Minimization SLAM to create dense

3D visual maps of underwater environments using a dense 3D stereo-vision system and IMU; it is an offline method. Shkurti et al. in [Shkurti et al. 2011] proposed a state estimation algorithm for underwater robot by combining information from monocular camera, IMU, and pressure sensor based on the multi-state constrained Kalman filter in [Mourikis and Roumeliotis 2007] .

Chapter 3

A Modular Sensor Suite for Underwater Reconstruction

3.1 Introduction

Localization and mapping in underwater environments is an important problem, common in many fields such as marine archeology, search and rescue, resource management, hydro-geology, and speleology. Target environments include, but are not limited to wrecks (ships/boats, planes, and buses), underwater structures (bridges, docks, and dams), and underwater caves in [Weidner et al. 2017; Mallios et al. 2017; Stone 2007; Gary et al. 2008]. Underwater environments present a huge challenge for vision-only mapping and navigation systems, making the deployment of autonomous underwater vehicles still an open problem. Light and color attenuation, due to the presence of particulates in the water, often combined with the complete absence of natural light, present major challenges. The combination of Visual and Inertial data has gain popularity with several proposed methods for fusing the two measurements in [Mourikis and Roumeliotis 2007; Jones and Soatto 2011; Kelly and Sukhatme 2011; Leutenegger et al. 2015]. In addition, most of the state-of-the-art visual or visual-inertial odometry algorithms have been shown to fail in underwater environments in [Quattrini Li et al. 2016]. However, vision still remains an accessible, easily interpretable sensor. On the other hand, the majority of underwater sensing for localization is based on acoustic sensors, such as ultrashort baseline (USBL) and Doppler Velocity Logger (DVL). Unfortunately, such sensors are usually expensive and could possibly disturb divers and/or the environment.

This paper presents the design, development, and deployment of an underwater sensor suite to be operated by human divers. The literature mainly focuses on AUVs and Autonomous Surface Vehicles (ASVs), and a body of work studies the Simultaneous Mapping and Localization (SLAM) problem and oceanographic reconstruction. Leedekerken et al. in [Leedekerken, Fallon, and Leonard 2014] presented an Autonomous Surface Craft (ASC) for concurrent mapping both above and below the water surface in large scale marine environments using a surface craft equipped

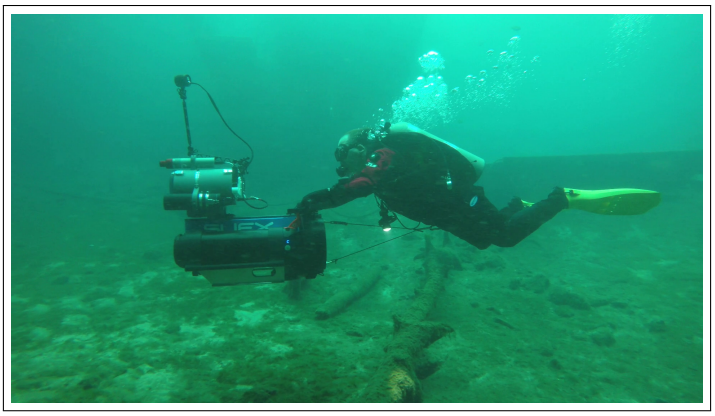


Figure 3.1: Our proposed underwater sensor suite mounted on a dual Diver Propulsion Vehicle (DPV), where a stability check was performed at Blue Grotto, FL.

with imaging sonar for subsurface perception and LIDAR, camera, and radar for perception above the surface.

Folaga in [Alvarez et al. 2005, a low cost AUV, can navigate on the sea surface and dive only at selected geographical points when measurements are needed. Roman et al. in [Roman et al. 2000] proposed an AUV equipped with camera and pencil beam sonar for applications including underwater photo-mosaicking, 3D image reconstruction, mapping, and navigation. AQUA in [Dudek et al. 2005], a visually guided legged swimming robot uses vision to navigate underwater and the target application areas are environmental assessment in [Hogue, German, and Jenkin 2007 and longitudinal analysis of coral reef environments in [Giguere et al. 2009]. Our aim is to accelerate state estimation research in the underwater domain that can be eventually deployed robustly in autonomous underwater vehicles (AUV) by enabling easy data collection by human divers. In particular, a specific target application is cave mapping, where the diving community has protocols in place for exploring and mapping such dangerous environments. The primary design goal of the proposed underwater sensor suite is to reduce the cognitive load of human divers by employing robotic technologies to map underwater structures. A second design goal is to enable software interoperability between different platforms, including AUVs. In particular,

the sensor suite presented in this paper contains identical sensors with an Aqua2 AUV in [Dudek et al. 2005], and can be deployed in different modes, hand-held by a diver, mounted on a single Diver Propulsion Vehicle (DPV), or on a dual DPV for better stability; see Fig. 3.1. The selected sensors include a mechanical scanning sonar, which provides robust range information about the presence of obstacles. Such a design choice improves the scale estimation by fusing acoustic range data into the visual-inertial framework in [Rahman, Quattrini Li, and Rekleitis 2018b].

The chapter is structured as follows. The next section outlines the design layout of hardware and software, deployment strategies, and the two versions of the sensor suite. The chapter concludes with a discussion on lessons learned and directions of future work.

3.2 Sensor Suite Design

The sensor suite hardware has been designed with underwater cave mapping in [Weidner et al. 2017 as the target application to be used by divers during cave exploration operations. In general, it can be used for mapping a variety of underwater structures and objects. In the following, the main requirements, hardware, and software components, are presented. Note that the full documentation for building and maintaining the hardware, as well as the necessary software can be found on our lab wiki page Autonomous Field Robotics Lab 2018.

3.2.1 Requirements

Given that the sensor suite will be primarily used by divers who are not necessarily engineers or computer scientists, the following requirements drive the hardware and software design of the proposed sensor suite:

- Portable.

- Neutrally buoyant.
- Hand-held or DPV deployment.
- Simple to operate.
- Waterproof to technical-diver operational depths.

Furthermore, the following desiderata are considered to make research in state estimation applied to the proposed sensor suite easily portable to other platforms, such as AUVs and ASVs:

- Standardization of hardware and software.
- Easy data storing.
- Low cost.

3.2.2 Hardware Design

In this section, the electronics selected and the designed enclosure are discussed, together with lessons learned during the construction of the proposed sensor suite.

Electronics

To assist vision-based state estimation, we employ an Inertial Measurement Unit (IMU), a depth, and an acoustic sensor for accurate state estimation in underwater environments. The specific sensors and electronics of the sensor suite were selected for compatibility with the Aqua2 Autonomous Underwater Vehicles (AUVs) in [Dudek et al. 2005. Figure 3.2 shows the computer and internal sensors on a Plexiglas plate, where the different electronic boards were placed optimizing the space to reduce the size of the sensor suite. In particular, the electronics consists of:

- two IDS UI-3251LE cameras in a stereo configuration,

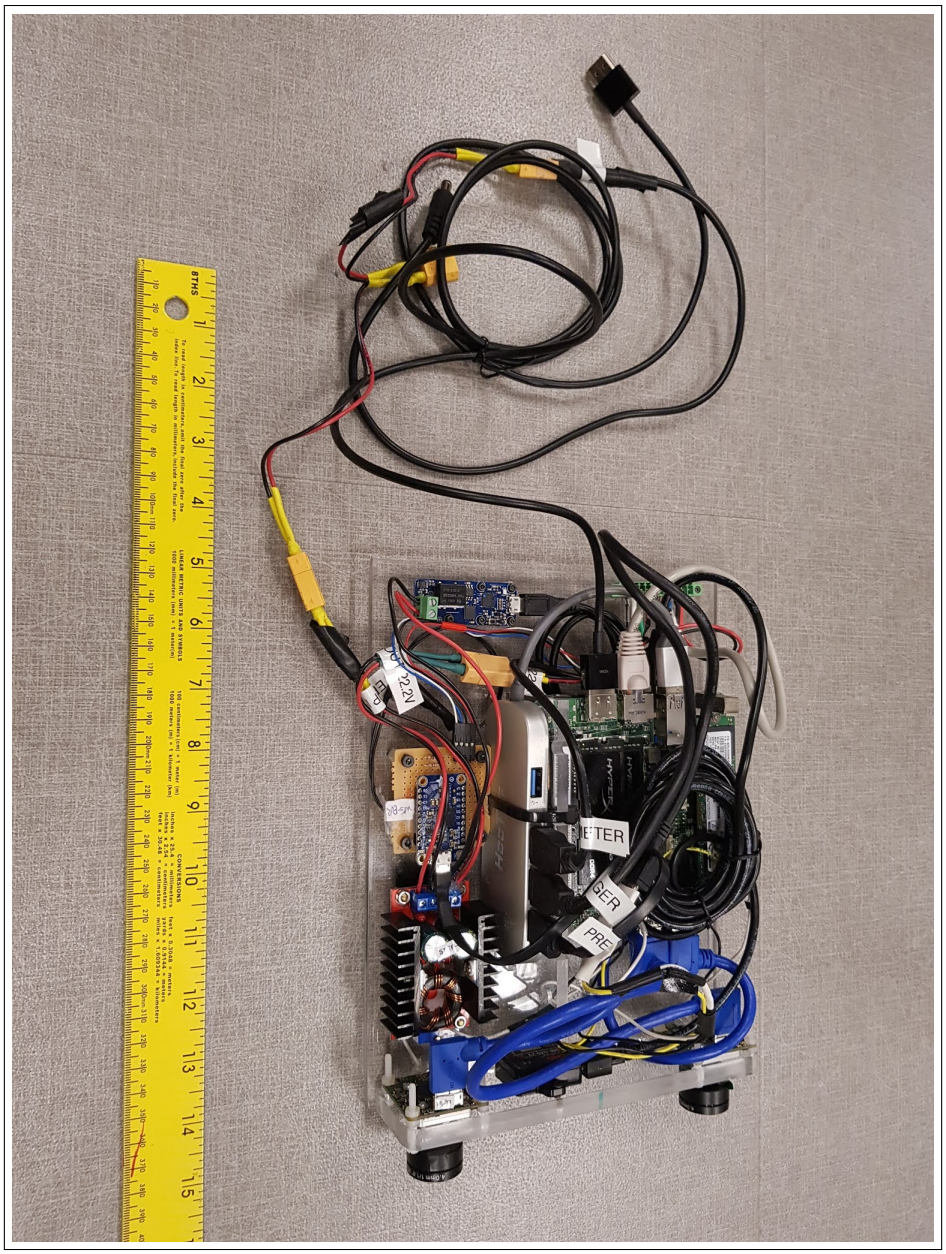
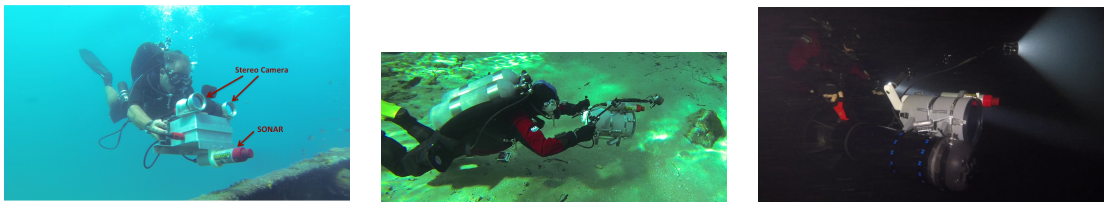


Figure 3.2: The Main Unit containing stereo camera, IMU, Intel NUC, and Pressure sensor.

- Microstrain 3DM-GX4-15 IMU,
- Bluerobotics Bar30 pressure sensor,
- Intel NUC as the computing unit,
- IMAGENEX 831L Sonar.



(a) (b) (c)

Figure 3.3: (a) First version of the stereo vision setup, where the two cameras are mounted externally to the main unit. (b) Second version of the sensor suite, where the stereo camera is inside the main unit. (c) Second version where the sensor suite is mounted on a DPV.

The two cameras are synchronized via a TinyLily, an Arduino-compatible board, and are capable of capturing images of 1600×1200 resolution at 20 Hz. The sonar provides range measurement with maximum range of 6 m distance, scanning in a plane over 360°, with angular resolution of 0.9°. A complete scan at 6 m takes 4 s. Note that the sonar provides for each measurement (point) 255 intensity values, that is 6/255 m is the distance between each returned intensity value. Clearly, higher response means a more likely presence of an obstacle. Sediment on the floor, porous material, and multiple reflections result in a multi-modal distribution of intensities. The IMU produces linear accelerations and angular velocities in three axis at a frequency of 100 Hz. Finally, the depth sensor produces depth measurements at 1 Hz. To enable the easy processing of data, the Robot Operating System (ROS) framework in [**ros** has been utilized for the sensor drivers and for recording timestamped data.

A 5 inch LED display has been added to provide visual feedback to the diver together with a system based on AR tags is used for changing parameters and to start/stop the recording underwater in [X. Wu et al. 2015 (see Section 3.2.3).

Enclosure

The enclosure for the electronics has been designed to ensure ease of operations by divers and waterproofness up to 100 m. In particular, two different designs were

tested. Both of them are characterized by the presence of handles for hand-held operations. The handles have been chosen so that a dive light can be easily added using a set of articulated arms. Note that all enclosures are sealed with proper o-rings/gaskets (details are reported in the linked documentation).

In the first design (see Fig. 3.3(a)) the main unit, a square shaped aluminum box – composed of two parts tighten together by screws – contained the computer, sensors, and other related electronics. The two cameras were sealed in aluminum tubes with tempered glass in front of the camera lenses. The stereo camera and display were mounted on the top of the main unit whereas the sonar was on the bottom of it. Both the cameras and sonar were connected to the main unit by underwater cables. The rationale behind such a design was to allow for an adjustable stereo baseline. Unfortunately, the USB 3.0 interfacing standard used by the cameras is not compatible with the underwater cables available in the market, resulting in highly degraded performance for the cameras with multiple dropped frames. In addition, the aluminum body made the sensor suite relatively heavy and negative buoyant. Furthermore, the position of the screen was not optimal for seeing it during regular diver deployment.

In the second design (see Fig. 3.3b), we took into account the lessons learned from the first design. In particular, a PVC tube was used instead of the aluminum box. This made the enclosure lighter and positive buoyant. Some rails at the bottom allows for additional weights for ballasting. Furthermore, the main enclosure hosted the two cameras as well. In this way, the cameras can be directly connected to the computer with standard USB 3.0 cables, to avoid unnecessary transmission of data over underwater cables as it was in the first design. The front panel is made of transparent Plexiglas, 33 mm thickness, while the back panel is made of aluminum, where a waterproof switch, a display, pressure sensor, and underwater connector for the sonar are mounted. Stainless steel Latches are used to close the panels with the

PVC tube, so that it can be easily open and maintained. The sonar was mounted on the top with the scanning plane parallel to the image plane and connected to the main unit by a standard SubConn underwater cable. Such a design and choice of material reduced the size and weight, and made it easier to carry and maintain. In addition, the second design of the sensor suite allows for modularity in terms of electronics used: a Plexiglas plate inside the enclosure was used to mount all the electronics and can be easily removed for troubleshooting or changed with a different computer, cameras, and IMU.

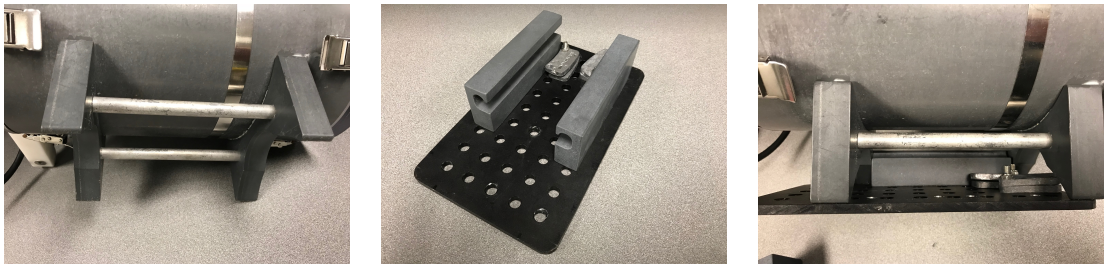
The second version of the sensor suite has been designed considering two different deployment strategies: hand-held and on different diver propulsion vehicles (DPV). Such deployment strategies depend on the structure of the environment and the distance to cover. The hand-held approach is more appropriate for covering a smaller area for a short period of time, whereas the sensor suite can be mounted on a single or double DPV in order to collect data over longer distances while being under water. Mounting the rig on a DPV is specifically useful in cave diving, at larger depths, to make better use of limited underwater time. Hand-held operations are possible through the handles on the side of the PVC tube, as shown in Fig. 3.3(b). DPV operations can be performed in two ways. First, mounted on a single DPV unit; see Fig. 3.3(c). Second, mounted on a dual DPV unit; see Fig. 3.1.

Fig. 3.4 shows a front view of the sensor suite fully assembled. The two side-ring holders are used to mount a canister battery for the video light; usually, a 13.5Ah NiMH standard battery.

Mounting Options

Mounting the sensor suite on single or dual DPVs uses different attachment methods. For single DPV attachment hose-clamps are used through the two metal bars to secure the sensor; see Fig. 3.5a. Please note, the bottom of the sensor suite has a

Figure 3.4: Front top view of the assembled sensor suite.



(a) (b) (c)

Figure 3.5: (a) The mounting system for single DPV deployment. (b) Mounting attachment for use with a dual DPV. (c) The dual DPV attachment partially mount on the bottom of the sensor suite.

round hollow that fits on a SUEX[1] DPV; either XJ37 or XK1 models. For mounting on a dual DPV, an attachment system is used; see Fig. 3.5b. The PVC components are hooked through the supporting metal poles at the bottom; see Fig. 3.5c where the plate is half mounted. When the plate is attached to the bottom of the sensor suite, then it locks on the railing system of the dual DPV unit. The mounting on the DPV can be carried out while in water, allowing divers to easily carry modular

[1] `https://www.suex.it/`

parts to the entry point for the dive. It is worth noting that the cheese-board and rail design allow for changing the location of the sensor on the dual DPV.

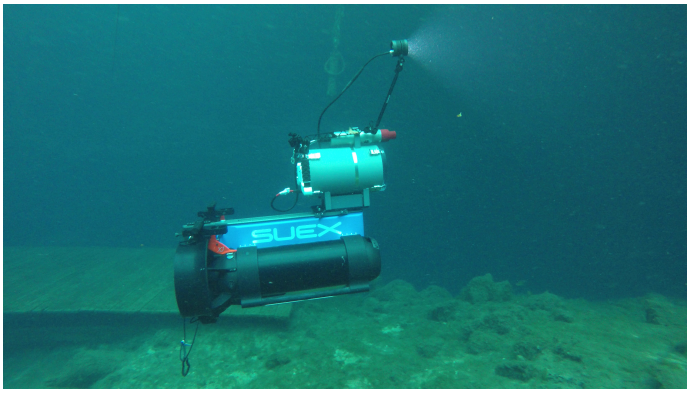


Figure 3.6: Sensor suite on a dual DPV free floating, neutrally buoyant.

Fig. 3.6 demonstrates the stability of the sensor suite on a dual DPV. The unit floats in the water neutrally buoyant, with the video light on top illuminating forward.

3.2.3 Software Design

The main software components of the sensor suite consist of:

- drivers for each hardware unit,
- a ROS interface for communication between sensors and data processing,
- an interface for user and sensor suite interaction.

Drivers

The aim for the software design is to have a modular system that ensures re-usability for both the system as a whole and also for each component. Each driver provides consistent interface for communication with the Robot Operating System (ROS) framework in [**ros** The main ROS drivers are:

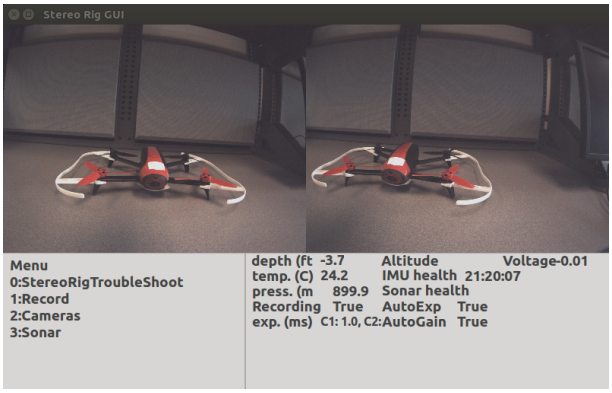


Figure 3.7: The default view of the menu.

- UEye driver for each camera, together with the Arduino code for the trigger to synchronize the cameras – available open-source [Anqi Xu and contributors 2018].
- IMU driver – available open-source [Kumar Robotics 2018].
- Sonar driver – developed in our lab, released open-source [Autonomous Field Robotics Lab 2018].
- Depth sensor driver - developed in our lab, released open-source [Autonomous Field Robotics Lab 2018].

ROS platform

For easy data collection, each sensor node publishes the related data. All the operations are performed on the computer that runs a Linux-based operating system. In particular, the Software was tested both on Ubuntu 14.04 and 16.04. After the operating system boot, a startup script runs all sensor nodes and at the same time starts the recording of sensor data through ROS bag file[2] that allows for easy play-back.

Interface

The interface consists of two components: Graphical User Interface (GUI) for online data monitoring; and AR tags [Fiala 2004] that supports user and sensor suite interaction, similarly to the proposed system by Sattar et al. [Sattar et al. 2007]. The GUI – based on Qt[3] for modularity – shows the current video stream of each camera and outputs the overall health of the system. Fig. 3.7 shows the sensor data from the GUI. Depth in feet and altitude represent the distance from the surface and from the bottom respectively; measured by the depth and the Sonar sensors. The temperature of the CPU is also reported in case there is overheating, especially if operations are started above water. In addition, the GUI shows a menu with a list of options that a user can select; left side of the screen. Each option has a corresponding AR tag associated with its number. Through the menu a user can perform basic operations on the computer – such as reboot or shutdown – start or stop recording data, get access to both camera or sonar settings. When a camera is selected, a user can change its gain and exposure and perform camera calibration. In addition, sonar data can be visualized through rviz[4] by selecting the corresponding option. Note that such a menu is modular and straightforward to add, remove, or modify the menu entries. Fig. 3.7 shows how the GUI looks like.

3.3 Conclusion

In this paper, we presented the design and development of a sensor suite for underwater reconstruction, together with some lessons learned during its construction. Our proposed sensor suite has been used by divers in coral reefs, shipwrecks, and cave

[2] `http://wiki.ros.org/rosbag`

[3] `https://www.qt.io/`

[4] `http://wiki.ros.org/rviz`

systems to collect visual, inertial, and sonar data, and different algorithms have been studied to improve state estimation in caves.

Immediate future work on the proposed sensor suite includes a comprehensive study on the quality of cameras for underwater operations, as well as a more user-friendly electronics placement and wiring. More broadly, such a sensor suite will be mounted on a platform that can operate autonomously, to allow for easy swap of sensors on a robot.

Chapter 4

An Underwater SLAM System using Sonar, Visual, Inertial, and Depth Sensor

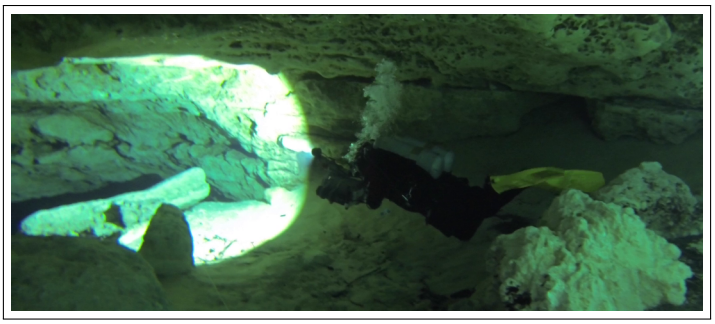


Figure 4.1: Underwater cave in Ginnie Springs, FL, where data have been collected using an underwater stereo rig.

4.1 Introduction

Most of the underwater navigation algorithms in [Leonard and Durrant-Whyte 2012; Lee et al. 2005; Snyder 2010; Johannsson et al. 2010; Rigby, Pizarro, and S. B. Williams 2006] are based on acoustic sensors, such as Doppler velocity log (DVL), Ultra-short Baseline (USBL), and sonar. However, data collection with these sensors is expensive and sometimes not suitable due to the highly unstructured underwater environments. In recent years, many vision-based state estimation algorithms have been developed using monocular, stereo, or multi-camera system mostly for indoor and outdoor environments. Vision is often combined with Inertial Measurement Unit (IMU) for improved estimation of pose in challenging environments, termed as *Visual-Inertial Odometry* (VIO) in [Mur-Artal and Juan D Tardós 2017; Leutenegger et al. 2015; Qin, Li, and Shen 2018; Mourikis and Roumeliotis 2007; Sun et al. 2018]. However, the underwater environment – e.g., see Fig. 4.1 – presents unique challenges to vision-based state estimation. As shown in a previous study in [Quattrini Li et al. 2016], it is not straightforward to deploy the available vision-based state estimation packages underwater. In particular, suspended particulates, blurriness, and light and color attenuation result in features that are not as clearly defined as above water. Consequently results from different vision-based state estimation packages show a significant number of outliers resulting in inaccurate estimate or even

complete tracking loss.

In this chapter, we propose a novel SLAM system specifically targeted for underwater environments – e.g., wrecks and underwater caves – and easily adaptable for different sensor configuration: acoustic (mechanical scanning profiling sonar), visual (stereo camera), inertial (linear accelerations and angular velocities), and depth data. This makes our system versatile and applicable on-board of different sensor suites and underwater vehicles.

The idea is that acoustic range data, though sparser, provide robust information about the presence of obstacles, where visual features reside; together with a more accurate estimate of scale. To fuse range data from sonar into the traditional VIO framework, we propose a new approach of taking a visual patch around each sonar point, and introduce extra constraints in the pose graph using the distance of the sonar point to the patch. The proposed method operates under the assumption that the visual-feature based patch is small enough and approximately coplanar with the sonar point. The resulting pose-graph consists of a combination of visual features and sonar features. In addition, we adopt the principle of *keyframe* based approaches to keep the graph sparse enough to enable real-time optimization. A particular challenge arises from the fact that the sonar features at an area are sensed after some time from the visual features due to the sensor suite configuration.

In our recent work, *SVIn* in [Rahman, Quattrini Li, and Rekleitis 2018b], acoustic, visual, and inertial data is fused together to map different underwater structures by augmenting the visual-inertial state estimation package OKVIS in [Leutenegger et al. 2015]. This improves the trajectory estimate especially when there is varying visibility underwater, as sonar provides robust information about the presence of obstacles with accurate scale. However, in long trajectories, drifts could accumulate resulting in an erroneous trajectory.

In [Rahman, Quattrini Li, and Rekleitis 2018c], we extend our work by including

an image enhancement technique targeted to the underwater domain, introducing depth measurements in the optimization process, loop-closure capabilities, and a more robust initialization. These additions enable the proposed approach to robustly and accurately estimate the sensor's trajectory, where every other approach has shown incorrect trajectories or loss of localization.

To validate our proposed approach, first, we assess the performance of the proposed loop-closing method, by comparing it to other state-of-the-art systems on the EuRoC micro-aerial vehicle public dataset in [Burri et al. 2016], disabling the fusion of sonar and depth measurements in our system. Second, we test the proposed full system on several different underwater datasets in a diverse set of conditions. More specifically, underwater data – consisting of visual, inertial, depth, and acoustic measurements – has been collected using a custom made sensor suite in [Rahman, Quattrini Li, and Rekleitis 2018a] from different locales; furthermore, data collected by an Aqua2 underwater vehicle in [Dudek et al. 2005] include visual, inertial, and depth measurements. The results on the underwater datasets illustrate the loss of tracking and/or failure to maintain consistent scale for other state-of-the-art systems while our proposed method maintains correct scale without diverging. Experimental data were collected from the Ginnie ballroom cavern at High Springs, in Florida; a submerged bus in North Carolina; a fake underwater cemetery in Lake Jocassee in South Carolina; and an artificial shipwreck in Barbados. In all cases a custom sensor suite employing a stereo camera, a mechanical scanning profiling sonar, and an IMU was used.

The paper is structured as follows. The next section discusses related work. Section 4.2 presents the mathematical formulation of the proposed system and describes the approach developed for image preprocessing, pose initialization, loop-closure, and relocalization. Section 4.3 presents results from a publicly available aerial dataset and a diverse set of challenging underwater environments. We conclude this paper with

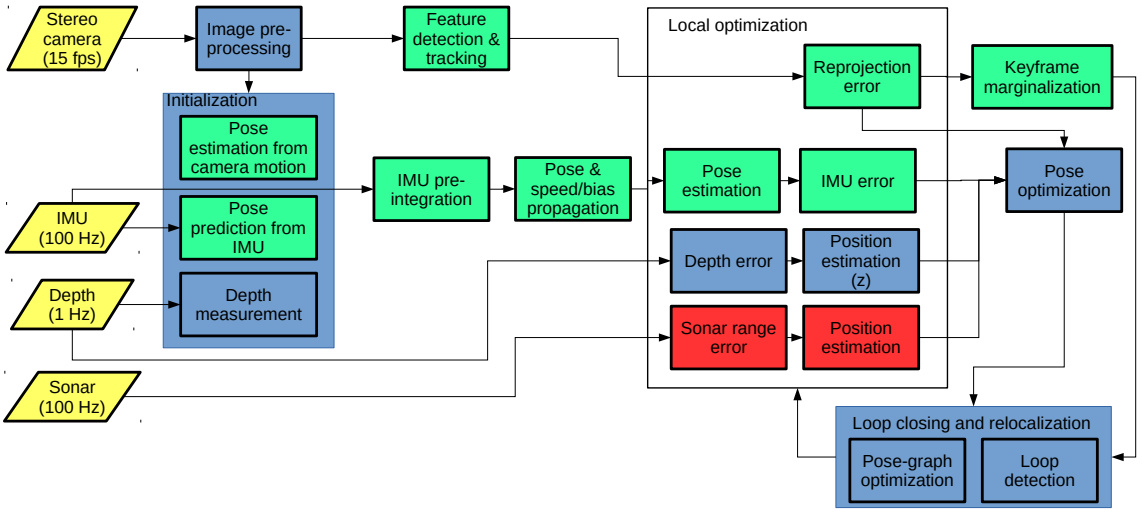


Figure 4.2: Block diagram of the proposed system, SVIn2; in yellow the sensor input, in green the components from OKVIS, in red the contribution from our work in [Rahman, Quattrini Li, and Rekleitis 2018b], and in blue the contributions in [Rahman, Quattrini Li, and Rekleitis 2018c].

a discussion on lessons learned and directions of future work.

4.2 Proposed Method

This section describes the proposed system, SVIn2, depicted in Fig. 4.2. The full proposed state estimation system can operate with a robot that has stereo camera, IMU, sonar, and depth sensor – the last two can be also disabled to operate as a visual-inertial system.

Due to low visibility and dynamic obstacles, it is hard to find good features to track. In addition to the underwater vision constraints, e.g., light and color attenuation, vision-based systems also suffer from poor contrast. Hence, we augment the pipeline by adding an image preprocessing step, where *contrast adjustment* along with *histogram equalization* is applied to improve feature detection underwater. In particular, we use a *contrast limited adaptive histogram equalization* filter in the *image pre-processing* step.

In the following, after defining the state, we describe the proposed initialization, sensor fusion optimization, loop closure and relocalization steps.

4.2.1 Notations and States

The full sensor suite is composed of the following coordinate frames: Camera (stereo), IMU, Sonar (acoustic), Depth, and World which are denoted as C, I, S, D, and W respectively. The transformation between two arbitrary coordinate frames X and Y is represented by a homogeneous transformation matrix ${}_X\mathbf{T}_Y = [{}_X\mathbf{R}_Y|{}_X\mathbf{p}_Y]$ where ${}_X\mathbf{R}_Y$ is rotation matrix with corresponding quaternion ${}_X\mathbf{q}_Y$ and ${}_X\mathbf{p}_Y$ is position vector.

Let us now define the robot R state $\mathbf{x}_R$ that the system is estimating as:

$$\mathbf{x}_R = [{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T, {}_W\mathbf{v}_I^T, \mathbf{b}_g{}^T, \mathbf{b}_a{}^T]^T \tag{4.1}$$

which contains the position ${}_W\mathbf{p}_I$, the attitude represented by the quaternion ${}_W\mathbf{q}_I$, the linear velocity ${}_W\mathbf{v}_I$, all expressed as the IMU reference frame I with respect to the world coordinate W; moreover, the state vector contains the gyroscopes and accelerometers bias $\mathbf{b}_g$ and $\mathbf{b}_a$.

The associated error-state vector is defined in minimal coordinates, while the perturbation takes place in the tangent space:

$$\delta\boldsymbol{\chi}_R = [\delta\mathbf{p}^T, \delta\mathbf{q}^T, \delta\mathbf{v}^T, \delta\mathbf{b}_g{}^T, \delta\mathbf{b}_a{}^T]^T \tag{4.2}$$

which represents the error for each component of the state vector with a transformation between tangent space and minimal coordinates in [Forster et al. 2017a].

4.2.2 Tightly-coupled Non-Linear Optimization with Sonar-Visual-Inertial-Depth (SVIND) measurements

For the tightly-coupled non-linear optimization, we use the following cost function $J(\mathbf{x})$, which includes the reprojection error $\mathbf{e}_r$ and the IMU error $\mathbf{e}_s$ with the addition

of the sonar error $\mathbf{e}_t$, and the depth error e_u:

$$\begin{aligned} J(\mathbf{x}) &= \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} \mathbf{e}_r^{i,j,k^T}\mathbf{P}_r^k\mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1}\mathbf{e}_s^{k^T}\mathbf{P}_s^k\mathbf{e}_s^k \\ &+ \sum_{k=1}^{K-1}\mathbf{e}_t^{k^T}\mathbf{P}_t^k\mathbf{e}_t^k + \sum_{k=1}^{K-1} e_u^{k^T}P_u^k e_u^k \end{aligned} \tag{4.3}$$

where i denotes the camera index – i.e., left ($i = 1$) or right ($i = 2$) camera in a stereo camera system with landmark index j observed in the k^{th} camera frame. $\mathbf{P}_r^k$, $\mathbf{P}_s^k$, $\mathbf{P}_t^k$, and P_u^k represent the information matrix of visual landmarks, IMU, sonar range, and depth measurement for the k^{th} frame respectively.

For completeness, we briefly discuss each error term – see [Leutenegger et al. 2015] for more details. The reprojection error describes the difference between a keypoint measurement in camera coordinate frame C and the corresponding landmark projection according to the stereo projection model. The IMU error term combines all accelerometer and gyroscope measurements by *IMU pre-integration* in [Forster et al. 2017a] between successive camera measurements and represents the *pose*, *speed and bias* error between the prediction based on previous and current states. Both reprojection error and IMU error term follow the formulation by Leutenegger in [Leutenegger et al. 2015].

The concept behind calculating the sonar range error is that, if the sonar detects any obstacle at some distance, it is more likely that the visual features would be located on the surface of that obstacle, and thus will be approximately at the same distance. The step involves computing a visual patch detected in close proximity of each sonar point to introduce an extra constraint, using the distance of the sonar point to the patch. Here, we assume that the visual-feature based patch is small enough and approximately coplanar with the sonar point.

In the presented system, the sonar measurements are used to correct the robot *pose* estimate as well as to optimize the use of landmarks coming from both vision and

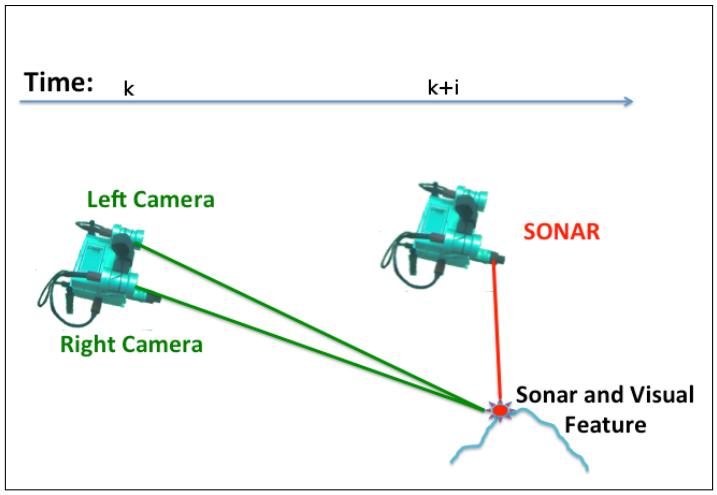


Figure 4.3: The relationship between sonar measurement and stereo camera features. A visual feature detected at time k is only detected by the sonar with a delay, at time $k+i$, where i depends on the speed the sensor is moving.

sonar. Due to the low visibility of underwater environments when it is hard to find visual features, sonar provides features with accurate scale. A particular challenge is the temporal displacement between the two sensors, vision and sonar. Figure 4.3 illustrates the structure of the problem, at time k some features are detected by the stereo camera, it takes some time (until $k+i$) for the sonar to pass by these visual features and thus obtain a related measurement. To address the above challenge, visual features detected in close proximity to the sonar return are grouped together and used to construct a patch. The distance between the sonar and the visual patch is used as an additional constraint.

For computational efficiency, the sonar range correction only takes place when a new camera frame is added to the pose graph. As sonar has a faster measurement rate than the camera, only the nearest *range* to the robot *pose* in terms of timestamp is used to calculate a small patch from *visual landmarks* around the sonar landmark for that given *range* and *head_position*. Algorithm 1 shows how to calculate the *range error* $\mathbf{e}_t^k$ given the robot position ${}_W\mathbf{p}_I^k$ and the sonar measurement $\mathbf{z}_t^k$ at time k.

The sonar returns *range* r and *head_position* θ measurements which are used to obtain each sonar landmark ${}_W\boldsymbol{l} = [l_x, l_y, l_z, 1]$ in homogeneous coordinate by a simple

Algorithm 1 SONAR Range Error Algorithm

Input: Estimation of robot position ${}_W\mathbf{p}_I^k$ at time k
Sonar measurement $\mathbf{z}_t^k = [r, \theta]$ at time k
List of current visual landmarks, $\mathcal{L}_v$
Distance threshold, T_d
Output: Range error e_t^k at time k
/*Compute sonar landmark in world coordinates*/
1: ${}_W\boldsymbol{l} = ({}_W\mathbf{T}_{II}\mathbf{T}_S[\mathbf{I}_3|r\cos(\theta), r\sin(\theta), 0]_S^T)$
/*Create list of visual landmarks around sonar landmark*/
2: $\mathcal{L}_S = \emptyset$
3: **for** (every $\boldsymbol{l}_i$ in $\mathcal{L}_v$) **do**
/*Compute Euclidean distance from visual landmark to sonar landmark*/
4: $d_S = \|{}_W l - l_i\|$
5: **if** ($d_S < T_d$) **then**
6: $\mathcal{L}_S = \mathcal{L}_S \cup \boldsymbol{l}_i$
7: **end if**
8: **end for**
9: $\hat{r} = \left\|{}_W\hat{\mathbf{p}}_I^k - \text{mean}(\mathcal{L}_S)\right\|$
10: **return** $r - \hat{r}$

geometric transformation in world coordinates:

$$ {}_W\boldsymbol{l} = ({}_W\mathbf{T}_{II}\mathbf{T}_S[\mathbf{I}_3|r\cos(\theta), r\sin(\theta), 0]_S^T) \tag{4.4} $$

where ${}_W\mathbf{T}_I$ and ${}_I\mathbf{T}_S$ are the respective transformation matrices used to transform the sonar measurement from the sonar coordinates to the world coordinates. ${}_I\mathbf{T}_S$ represents the transformation from the sonar frame of reference to the IMU reference frame, and ${}_W\mathbf{T}_I$ represents the transformation from the inertial (IMU) frame to the world coordinates. Consequently, the sonar range prediction is calculated using the lines 2-9 of Algorithm 1:

$$ \hat{r} = \|{}_W\hat{\mathbf{p}}_I - \text{mean}(\mathcal{L}_S)\|, \tag{4.5} $$

where $\mathcal{L}_S$ is the subset of visual landmarks around the sonar landmark. As mentioned above, the concept behind calculating the *range error* is that, if the sonar detects any obstacles at some distance, it is more likely that the visual features would be located on the surface of that obstacle, and thus will be at approximately the same

distance. Thus, the error term is the difference between the two distances. Note that we approximate the visual patch with the centroid (mean($\mathcal{L}_S$)), to filter out noise on the visual landmarks.

As such, given the sonar measurement $\mathbf{z}_t^k$, the error term $\mathbf{e}_t^k({}_W\mathbf{p}_I^k, \mathbf{z}_t^k)$ is based on the difference between those two distances which is used to correct the position ${}_W\mathbf{p}_I^k$. We assume an approximate normal conditional probability density function f with zero *mean* and $\mathbf{W}_t^k$ *variance*, and the conditional covariance $\mathbf{Q}(\delta\hat{\mathbf{p}^k}|\mathbf{z}_t^k)$, updated iteratively as new sensor measurements are integrated:

$$f(\mathbf{e}_t^k|{}_W\mathbf{p}_I^k) \approx \mathcal{N}(\mathbf{0}, \mathbf{W}_t^k) \tag{4.6}$$

The information matrix is:

$$\mathbf{P}_t^k = \mathbf{W}_t^{k^{-1}} = \left(\frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}^k}} \mathbf{Q}(\delta\hat{\mathbf{p}^k}|\mathbf{z}_t^k) \frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}^k}}^T \right)^{-1} \tag{4.7}$$

The Jacobian can be derived by differentiating the expected *range* r measurement with respect to the robot position:

$$\frac{\partial \mathbf{e}_t^k}{\partial \delta\hat{\mathbf{p}^k}} = \left[\frac{-l_x + {}_Wp_x}{r}, \frac{-l_y + {}_Wp_y}{r}, \frac{-l_z + {}_Wp_z}{r} \right] \tag{4.8}$$

The pressure sensor, introduced in this paper, provides accurate depth measurements based on water pressure. Depth values are extracted along the *gravity* direction which is aligned with the z of the world W – observable due to the tightly coupled IMU integration. The depth data at time k is given by[1]:

$${}_Wp_{z_D}{}^k = d^k - d^0 \tag{4.9}$$

With depth measurement z_u^k, the depth error term $e_u^k({}_Wp_{z_I}{}^k, z_u^k)$ can be calculated as the difference between the robot position along the z direction and the depth data

[1]More precisely, ${}_Wp_{z_D}{}^k = (d^k - d^0) + \textit{init\_disp\_from\_IMU}$ to account for the initial displacement along z axis from IMU, which is the main reference frame used by visual SLAM to track the sensor suite/robot.

to correct the position of the robot. The error term can be defined as:

$$e_u^k({}_W p_{z_I}{}^k, z_u^k) = |{}_W p_{z_I}^k - {}_W\, p_{z_D}^k| \tag{4.10}$$

The information matrix calculation follows a similar approach as the sonar and the Jacobian is straight-forward to derive.

All the error terms are added in the *Ceres Solver* nonlinear optimization framework in [Agarwal, Mierle, et al. 2015] to estimate the robot state.

4.2.3 Initialization: Two-step Scale Refinement

A robust and accurate initialization is required for the success of tightly-coupled non-linear systems, as described in in [Mur-Artal and Juan D Tardós 2017] and in [Qin, Li, and Shen 2018]. For underwater deployments, this becomes even more important as vision is often occluded as well as is negatively affected by the lack of features for tracking. Indeed, from our comparative study of visual-inertial based state estimation systems in [Joshi et al. 2019], in underwater datasets, most of the state-of-the-art systems either fail to initialize or make wrong initialization resulting into divergence. Hence, we propose a robust initialization method using the sensory information from stereo camera, IMU, and depth for underwater state estimation. The reason behind using all these three sensors is to introduce constraints on *scale* to have a more accurate estimation on initialization. Note that no acoustic measurements have been used because the sonar range and visual features contain a temporal difference, which would not allow to have any match between acoustic and visual features, if the robot is not moving. This is due to the fact that the sonar scans on a plane over 360° *around the robot* and camera detects features *in front of the robot* Rahman, Quattrini Li, and Rekleitis 2018b]; see Fig. 4.4.

In particular, the proposed initialization works as follows. First, we make sure that the system only initializes when a minimum number of visual features are present

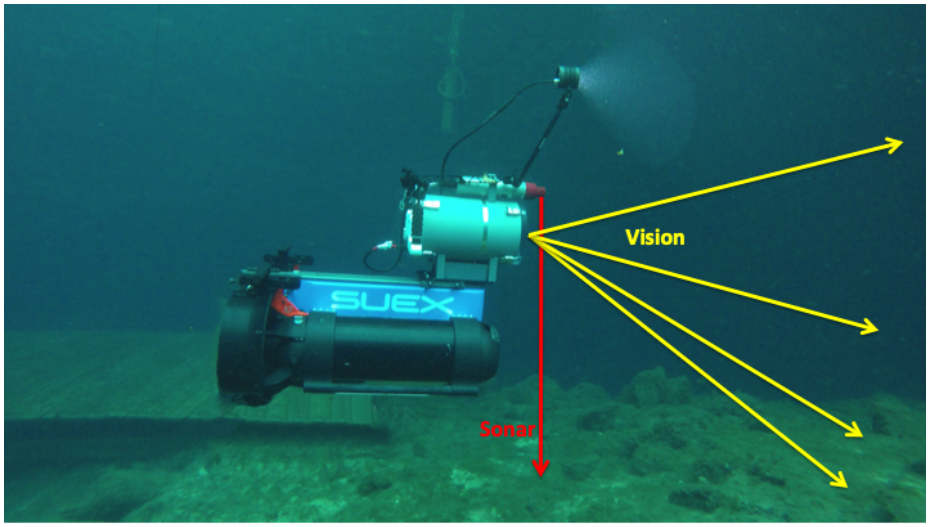


Figure 4.4: Custom made sensor suite mounted on a dual DPV. Sonar scans around the sensor while the cameras see in front.

to track (in our experiments 15 worked well). Second, the two-step refinement of the initial scale from the stereo vision takes place.

The depth sensor provides accurate depth measurements which are used to refine the initial scale factor from stereo camera. Including a scale factor s_1, the transformation between camera C and depth sensor D can be expressed as

$$ {}_W p_{z_D} = s_1 * {}_W p_{z_C} + {}_W\mathbf{R}_{z_{CC}} \mathbf{p}_D \tag{4.11} $$

For *keyframe* k, solving the above equation for s_1, provides the first refinement r_1 of the initial stereo scale ${}_W\mathbf{p}_{r1C}$, i.e.,

$$ {}_W\mathbf{p}_{r1C} = s_1 * {}_W\mathbf{p}_C \tag{4.12} $$

In the second step, the refined measurement from stereo camera in Eq. (4.12) is aligned with the IMU pre-integral values. Similarly, the transformation between camera C and IMU I with scale factor s_2 can be expressed as:

$$ {}_W\mathbf{p}_I = s_2 * {}_W\mathbf{p}_{r1C} + {}_W\mathbf{R}_{CC} \mathbf{p}_I \tag{4.13} $$

In addition to refining the scale, we also approximate initial *velocity* and *gravity* vector similar to the method described in in [Qin, Li, and Shen 2018]. The state prediction from IMU integration $\hat{\mathbf{x}}_R^{i+1}(\mathbf{x}_R^i, \mathbf{z}_I^i)$ with IMU measurements $\mathbf{z}_I^i$ in OKVIS in [Leutenegger et al. 2015 with conditional covariance $\mathbf{Q}(\delta\hat{\mathbf{x}}_R^{i+1}|\mathbf{x}_R^i, \mathbf{z}_I^i)$ can be written as (the details about IMU pre-integration can be found in in [Forster et al. 2017a]):

$$\begin{aligned} {}_W\hat{\mathbf{p}}_I^{i+1} &= {}_W\mathbf{p}_I^i + {}_W\mathbf{v}_I^i\Delta t_i + \frac{1}{2}{}_W\mathbf{g}\Delta {t_i}^2 + {}_W\mathbf{R}_I^i\boldsymbol{\alpha}_{I_i}^{i+1} \\ {}_W\hat{\mathbf{v}}_I^{i+1} &= {}_W\mathbf{v}_I^i + {}_W\mathbf{g}\Delta t_i + {}_W\mathbf{R}_I^i\boldsymbol{\beta}_{I_i}^{i+1} \\ {}_W\hat{\mathbf{q}}_I^{i+1} &= \boldsymbol{\gamma}_{I_i}^{i+1} \end{aligned} \tag{4.14}$$

where $\boldsymbol{\alpha}_{I_i}^{i+1}$, $\boldsymbol{\beta}_{I_i}^{i+1}$, and $\boldsymbol{\gamma}_{I_i}^{i+1}$ are IMU pre-integration terms defining the motion between two consecutive keyframes *i* and *i+1* in time interval Δt_i and can be obtained only from the IMU measurements. Eq. (4.14) can be re-arranged with respect to $\boldsymbol{\alpha}_{I_i}^{i+1}$, $\boldsymbol{\beta}_{I_i}^{i+1}$ as follows:

$$\begin{aligned} \boldsymbol{\alpha}_{I_i}^{i+1} &= {}_I\mathbf{R}_W^i({}_W\hat{\mathbf{p}}_I^{i+1} - {}_W\mathbf{p}_I^i - {}_W\mathbf{v}_I^i\Delta t_i - \frac{1}{2}{}_W\mathbf{g}\Delta {t_i}^2) \\ \boldsymbol{\beta}_{I_i}^{i+1} &= {}_I\mathbf{R}_W^i({}_W\hat{\mathbf{v}}_I^{i+1} - {}_W\mathbf{v}_I^i - {}_W\mathbf{g}\Delta t_i) \end{aligned} \tag{4.15}$$

Substituting Eq. (4.13) into Eq. (4.15), we can estimate $\boldsymbol{\chi}_S = [\mathbf{v}_I^i, \mathbf{v}_I^{i+1}, {}_W\mathbf{g}, s_2]^T$ by solving the linear least square problem in the following form:

$$\min_{\boldsymbol{\chi}_S} \sum_{i\in K} \left\| \hat{\boldsymbol{z}}_{S_i}^{i+1} - \mathbf{H}_{S_i}^{i+1}\boldsymbol{\chi}_S \right\|^2 \tag{4.16}$$

where $\hat{\boldsymbol{z}}_{S_i}^{i+1} =$

$$\begin{bmatrix} \hat{\boldsymbol{\alpha}}_{I_i}^{i+1} - {}_I\mathbf{R}_W^i{}_W\mathbf{R}_C^{i+1}{}_C\mathbf{p}_I^{i+1} + {}_I\mathbf{R}_C^i{}_C\mathbf{p}_I^i \\ \hat{\boldsymbol{\beta}}_{I_i}^{i+1} \end{bmatrix}$$

and $\mathbf{H}_{S_i}^{i+1} =$

$$\begin{bmatrix} -\mathbf{I}\Delta t_i & \mathbf{0} & -\frac{1}{2}{}_I\mathbf{R}_W^i\Delta {t_i}^2 & {}_I\mathbf{R}_W^i({}_W\mathbf{p}_{r1C}^{\ i+1} - {}_W\mathbf{p}_{r1C}^{\ i}) \\ -\mathbf{I} & {}_I\mathbf{R}_W^i{}_W\mathbf{R}_I^{i+1} & -{}_I\mathbf{R}_W^i\Delta t_i & \mathbf{0} \end{bmatrix}$$

4.2.4 Loop-closing and Relocalization

In a sliding window and marginalization based optimization method, drift accumulates over time on the pose estimate. A global optimization and relocalization scheme is necessary to eliminate this drift and to achieve global consistency. We adapt DBoW2 in [Gálvez-López and Tardos 2012], a bag of binary words (BoW) place recognition module, and augment OKVIS for loop detection and relocalization. For each keyframe, the descriptors for only the *keypoints* detected during the local tracking are used to build BoW database. No new features will be detected in the loop closure step.

A pose-graph is maintained to represent the connection between keyframes. In particular, a node represents a keyframe and an edge between two keyframes exists if the matched keypoints ratio between them is more than 0.75. In practice, this results into a very sparse graph. With each new keyframe in the pose-graph, the loop-closing module searches for candidates in the bag of words database. A query for detecting loops to the BoW database only returns the candidates outside the current marginalization window and having greater than or equal to score than the neighbor keyframes of that node in the pose-graph. If loop is detected, the candidate with the highest score is retained and feature correspondences between the current keyframe in the local window and the loop candidate keyframe are obtained to establish connection between them. The pose-graph is consequently updated with loop information. A 2D-2D descriptor matching and a 3D-2D matching between the known landmark in the current window keyframe and loop candidate with outlier rejection by PnP RANSAC is performed to obtain the geometric validation.

When a loop is detected, the global relocalization module aligns the current keyframe pose in the local window with the pose of the loop keyframe in the pose-graph by sending back the drift in pose to the windowed sonar-visual-inertial-depth optimization thread. Also, an additional optimization step, similar to Eq. (4.3), is

taken only with the matched landmarks with loop candidate for calculating the sonar error term and reprojection error; see Eq. (4.17).

$$J(\mathbf{x}) = \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in Loop(i,k)} \mathbf{e}_r^{i,j,k^T}\mathbf{P}_r^k\mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1}\mathbf{e}_t^{k^T}\mathbf{P}_t^k\mathbf{e}_t^k \tag{4.17}$$

After loop detection, a 6-DoF (position, $\mathbf{x_p}$ and rotation, $\mathbf{x_q}$) pose-graph optimization takes place to optimize over relative constraints between poses to correct drift. The relative transformation between two poses $\mathbf{T}_i$ and $\mathbf{T}_j$ for current keyframe in the current window i and keyframe j (either loop candidate keyframe or connected keyframe) can be calculated from $\Delta\mathbf{T}_{ij} = \mathbf{T}_j{\mathbf{T}_i}^{-1}$. The error term, $\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}$ between keyframes i and j is formulated minimally in the tangent space:

$$\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} = \Delta\mathbf{T}_{ij}\hat{\mathbf{T}}_i{\hat{\mathbf{T}}_j}^{-1} \tag{4.18}$$

where $(\hat{.})$ denotes the estimated values obtained from local sonar-visual-inertial-depth optimization. and the cost function to minimize is given by

$$J(\mathbf{x_p},\mathbf{x_q}) = \sum_{i,j}{\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}}^T\mathbf{P}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}} + \sum_{(i,j)\in Loop}\rho({\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}}^T\mathbf{P}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}\mathbf{e}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}) \tag{4.19}$$

where $\mathbf{P}^{i,j}_{\mathbf{x_p},\mathbf{x_q}}$ is the information matrix set to identity, as in in [Strasdat 2012], and ρ is the Huber loss function to potentially down-weight any incorrect loops.

4.3 Experimental Results

The proposed state estimation system, SVIn2, is quantitatively validated first on a standard dataset, to ensure that loop closure and the initialization work also above water. Moreover, it is compared to other state-of-the-art methods, i.e., VINS-Mono in [Qin, Li, and Shen 2018], the basic OKVIS in [Leutenegger et al. 2015], and the MSCKF in [Mourikis and Roumeliotis 2007] implementation from the GRASP lab

in [Research group of Prof. Kostas Daniilidis 2018]. Second, we qualitatively test the proposed approach on several different datasets collected utilizing a custom made sensor suite in [Rahman, Quattrini Li, and Rekleitis 2018a] and an Aqua2 AUV in [Dudek et al. 2005].

4.3.1 Validation on Standard dataset

Here, we present results on the EuRoC dataset in [Burri et al. 2016], one of the benchmark datasets used by many visual-inertial state estimation systems, including OKVIS (Stereo), VINS-Mono, and MSCKF. To compare the performance, we disable depth and sonar integration in our method and only assess the loop-closure scheme.

Following the current benchmarking practices, an alignment is performed between ground truth and estimated trajectory, by minimizing the least mean square errors between estimate/ground-truth locations, which are temporally close, varying rotation and translation, according to the method from Umeyama 1991]. The resulting metric is the Root Mean Square Error (RMSE) for the translation, shown in Table 4.1 for several Machine Hall sequences in the EuRoC dataset. For each package, every sequence has been run 5 times and the best run (according to RMSE) has been shown. Our method shows reduced RMSE in every sequence from OKVIS, validating the improvement of pose-estimation after loop-closing. SVIn2 has also less RMSE than MSCKF and slightly higher in some sequences, but comparable, to results from VINS-Mono. Fig. 4.5 shows the trajectories for each method together with the ground truth for the *Machine Hall* sequence.

4.3.2 Underwater datasets

Our proposed state estimation system – SVIn2 – is targeted for the underwater environment, where sonar and depth can be fused together with the visual-inertial data. Here, we show results from four different datasets in three different underwater en-

Table 4.1: The best absolute trajectory error (RMSE) in meters for each Machine Hall EuRoC sequence.

	SVIn2	OKVIS(stereo)	VINS-Mono	MSCKF
MH 01	0.13	0.15	0.07	0.21
MH 02	0.08	0.14	0.08	0.24
MH 03	0.07	0.12	0.05	0.24
MH 04	0.13	0.18	0.15	0.46
MH 05	0.15	0.24	0.11	0.54

vironments. First, a sunken bus in Fantasy Lake (NC), where data was collected by a diver with a custom-made underwater sensor suite [Rahman, Quattrini Li, and Rekleitis 2018a]. The diver started from outside the bus, performed a loop around and entered in it from the back door, exited across and finished at the front-top of the bus. The images are affected by haze and low visibility. Second and third, data from an underwater cavern in Ginnie Springs (FL) is collected again by a diver with the same sensor suite as for the sunken bus. The diver performed several loops, around one spot in the second dataset – Cavern1 – and two spots in the third dataset – Cavern2 – inside the cavern. The environment is affected by complete absence of natural light. Fourth, an AUV – Aqua2 robot – collected data over a fake underwater cemetery in Lake Jocassee (SC) and performed several loops around the tombstones in a square pattern. The visibility, as well as brightness and contrast, was very low. In the underwater datasets, it is a challenge to get any ground truth, because it is a GPS-denied unstructured environment. As such, the evaluation is qualitative, with a rough estimate on the size of the environment measured beforehand by the divers collecting the data.

Figs. 4.7-4.10 show the trajectories from SVIn2, OKVIS, and VINS-Mono in the

datasets just described. MSCKF was able to keep track only for some small segments in all the datasets, hence excluded from the plots. For a fair comparison, when the trajectories were compared against each other, sonar and depth were disabled in SVIn2. All trajectories are plotted keeping the original scale produced by each package.

Fig. 4.7 shows the results for the submerged bus dataset. In particular, VINS-Mono lost track when the exposure increased for quite some time. It tried to re-initialize, but it was not able to track successfully. Even using *histogram equalization* or a *contrast adjusted histogram equalization* filter, VINS-Mono was not able to track. Even if the scale drifted, OKVIS was able to track using a *contrast adjusted histogram equalization* filter in the image pre-processing step. Without the filter, it lost track at the high exposure location. The proposed method was able to track, detect, and correct the loop, successfully.

In Cavern1 – see Fig. 4.8 – VINS-Mono tracked successfully the whole time. However, as can be noticed in Fig. 4.8c, the scale was incorrect based on empirical observations during data collection. OKVIS instead produced a good trajectory, and SVIn2 was also able to detect and close the loops.

In Cavern2 (Fig. 4.9), VINS-Mono lost track at the beginning, reinitialized, was able to track for some time, and detected a loop, before losing track again. VINS-Mono had similar behavior even if the images were pre-processed with different filters. OKVIS tracked well, but as drifts accumulated over time, it was not able to join the current pose with a previous pose where a loop was expected. SVIn2 was able to track and reduce the drift in the trajectory with successful loop closure.

In the cemetery dataset – Fig. 4.10 – both VINS-Mono and OKVIS were able to track, but VINS-Mono was not able to reduce the drift in trajectory, while SVIn2 was able to fuse and correct the loops.

4.3.3 Reconstruction using Sonar data

The proposed approach has been tested in numerous challenging environments. In the following a description of each dataset together with the state estimate of the sensor suite and challenges during the field trials are presented.

One of the first datasets was collected at an artificial shipwreck in Barbados; see Fig. 4.12a. The initial deployment of the sonar sensor suffered from a misconfiguration where data was collected at a very slow rate and at a maximum range of one meter resulting on only collecting sonar data from the floor. Note that Fig. 4.12c shows the trajectory of the camera going slightly upwards, although the frame shows the floor parallel to the motion. The shipwreck was sunken on the sea floor with some inclination, that the IMU was able to capture.

We collected also a short segment from inside a cavern in Ginnie Springs, in Florida (USA). Such footage provided preliminary data from an underwater cave environment; see Fig. 4.13a. The video light utilized was providing illumination on only part of the scene. The reconstruction shows both visual landmarks and sonar points giving a sense of the cavern as the diver was swimming around. In such a case, the sonar was correctly configured; however, because the light was not uniformly illuminating the scene, the visual features were not optimal.

Finally, the inside of a sunken bus was mapped at Fantasy Lake Scuba Park, NC, USA; see Fig. 4.14a. The image quality was quite poor due to the many particulates in the water. In all environments, the images contain a significant amount of blur (softness) which clearly increases with distance. In addition, dynamic obstacles, such as fish, but more importantly floating particles that reflect back with high intensity; see Fig. 4.11, where floating particles were present in all datasets.

In such challenging environments, it is very hard to get ground truth. However, the trajectory and the distance covered resembled the one followed by the diver. Further, the sonar landmarks were indeed used to correct the pose estimate. All the

results in the datasets, but the shipwreck, show several rings, indicating the mapping of the structure.

4.4 Conclusions

In this chapter, we presented a state estimation system with robust initialization, sensor fusion of depth, sonar, visual, and inertial data, and loop closure capabilities. While the proposed system can also work out of the water, by disabling the sensors that are not applicable, our system is specifically targeted for underwater environments. Experimental results in a standard benchmark dataset and different underwater datasets demonstrate excellent performance.

Utilizing the insights gained from implementing the proposed approach, an online adaptation of the discussed framework for the limited computational resources of the Aqua2 AUV in [Dudek et al. 2005] is currently under consideration; see Fig. 4.6. It is worth noting that maintaining the proper attitude of the traversed trajectory and providing an estimate of the distance traveled will greatly enhance the autonomous capabilities of the vehicle in [Sattar et al. 2008]. Furthermore, accurately modeling the surrounding structures would enable Aqua2, as well as other vision based underwater vehicles to operate near, and through, a variety of underwater structures, such as caves, shipwrecks, and canyons.

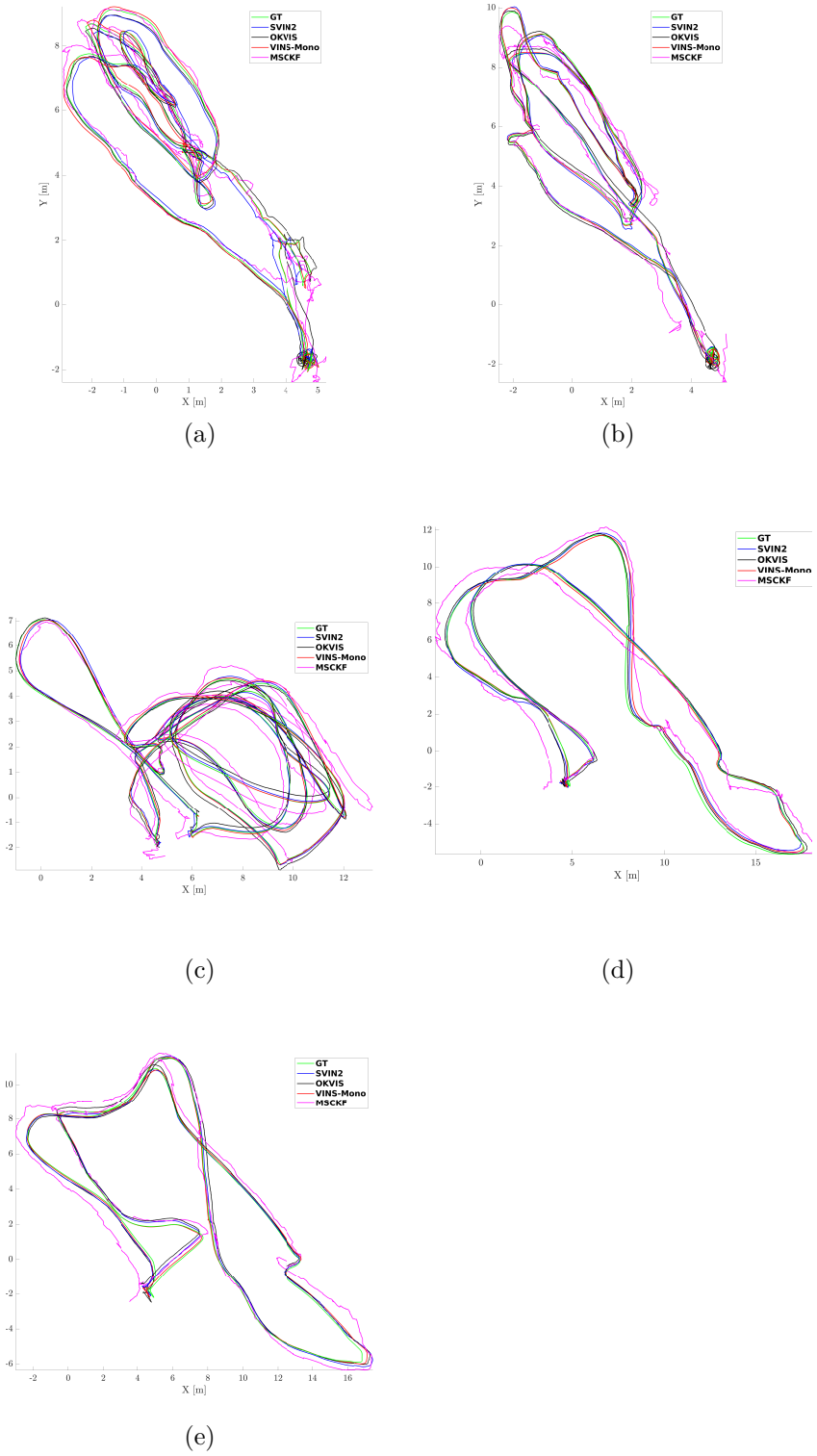


Figure 4.5: Trajectories on the MH sequence of the EuRoC dataset.

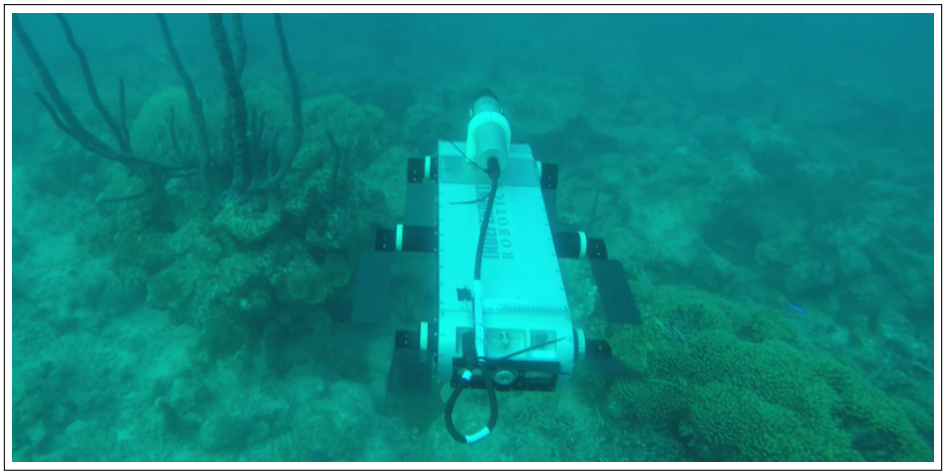


Figure 4.6: The Aqua2 AUV in [Dudek et al. 2005] equipped with the scanning sonar collecting data over the coral reef.

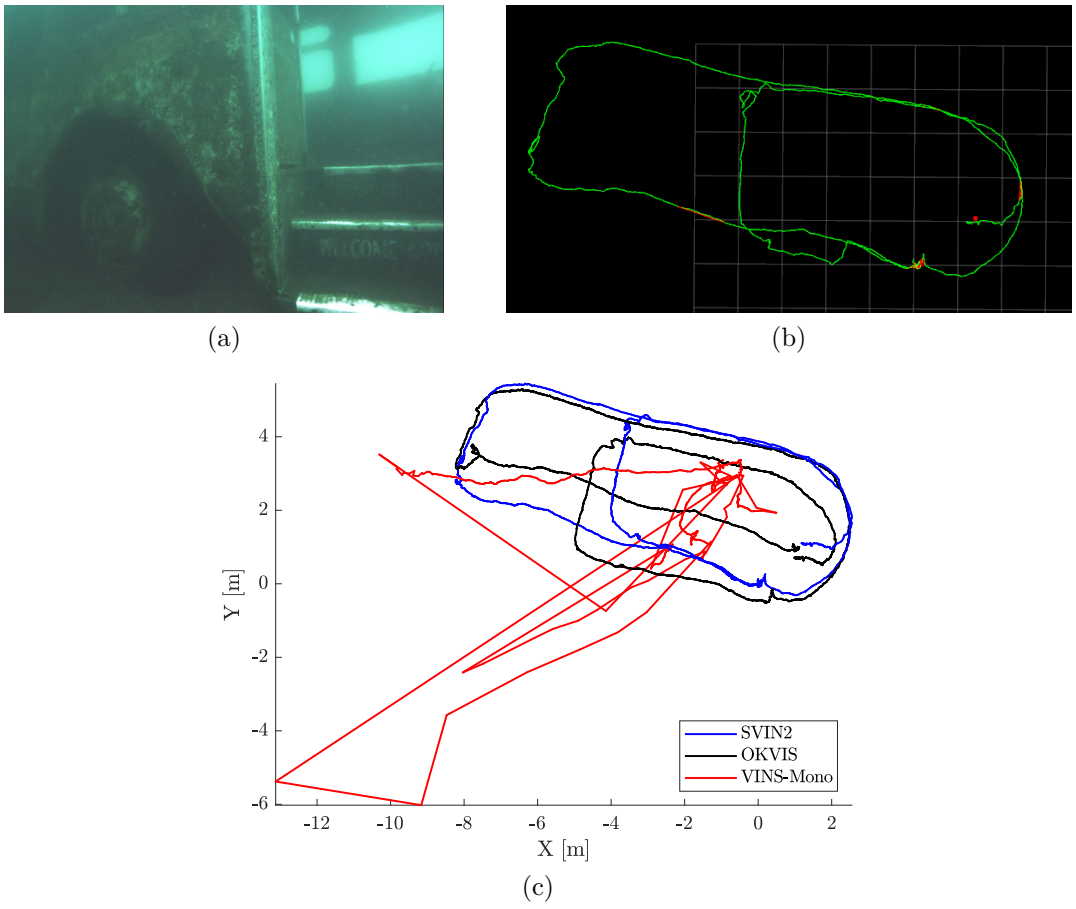
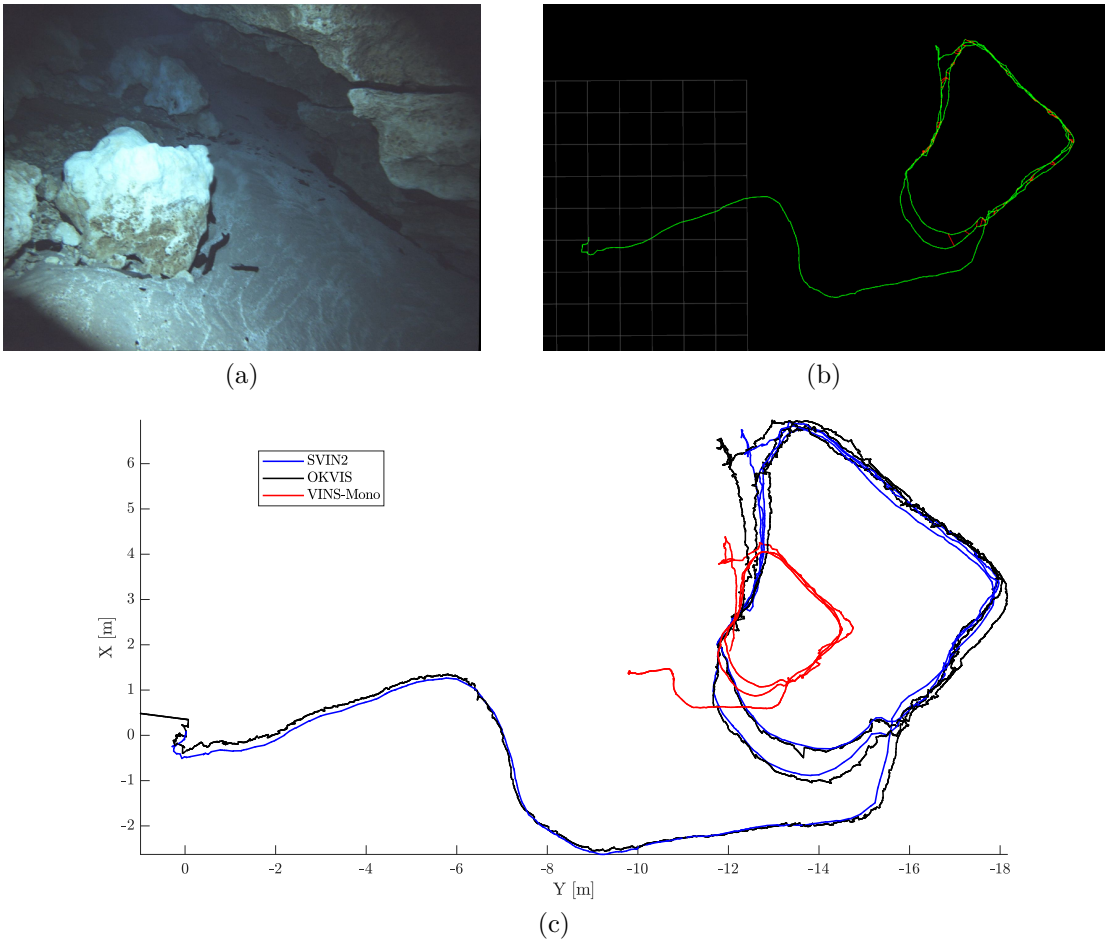
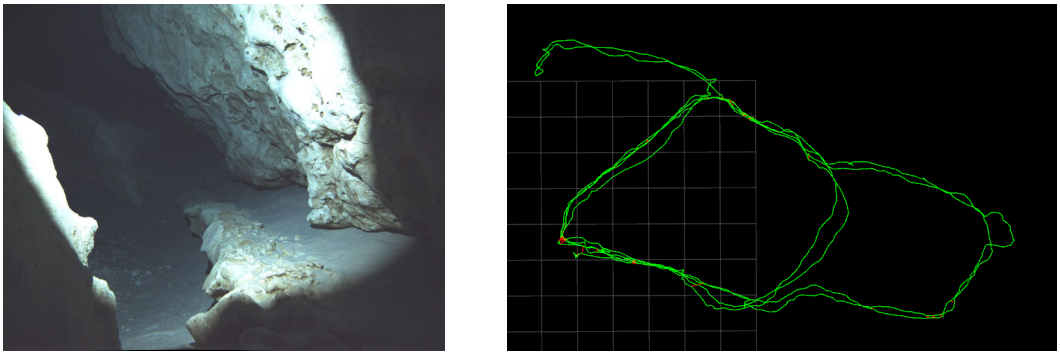


Figure 4.7: (a) Submerged bus, Fantasy Lake, NC, USA; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

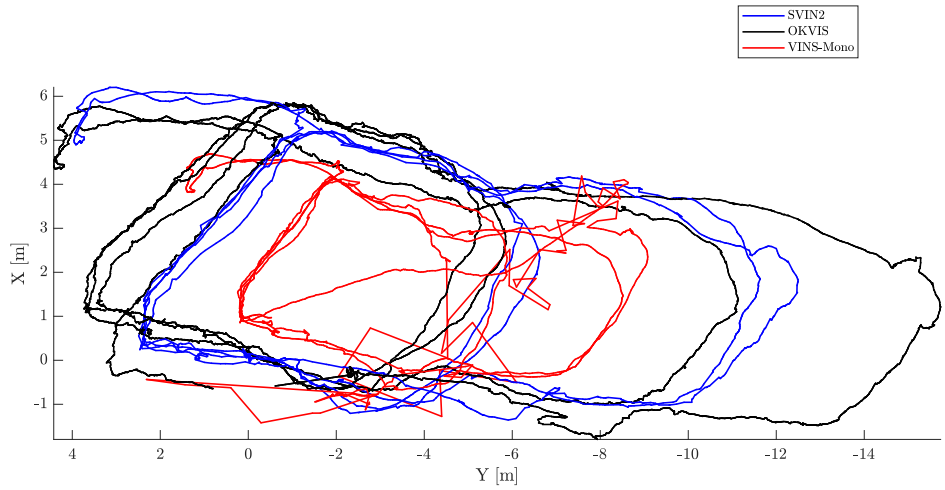


(a) (b) (c)

Figure 4.8: (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with a unique loop; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.



(a) (b)



(c)

Figure 4.9: (a) Cave environment, Ballroom, Ginnie Springs, FL, USA, with two loops in different areas; trajectories from SVIn2 with all sensors enabled shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

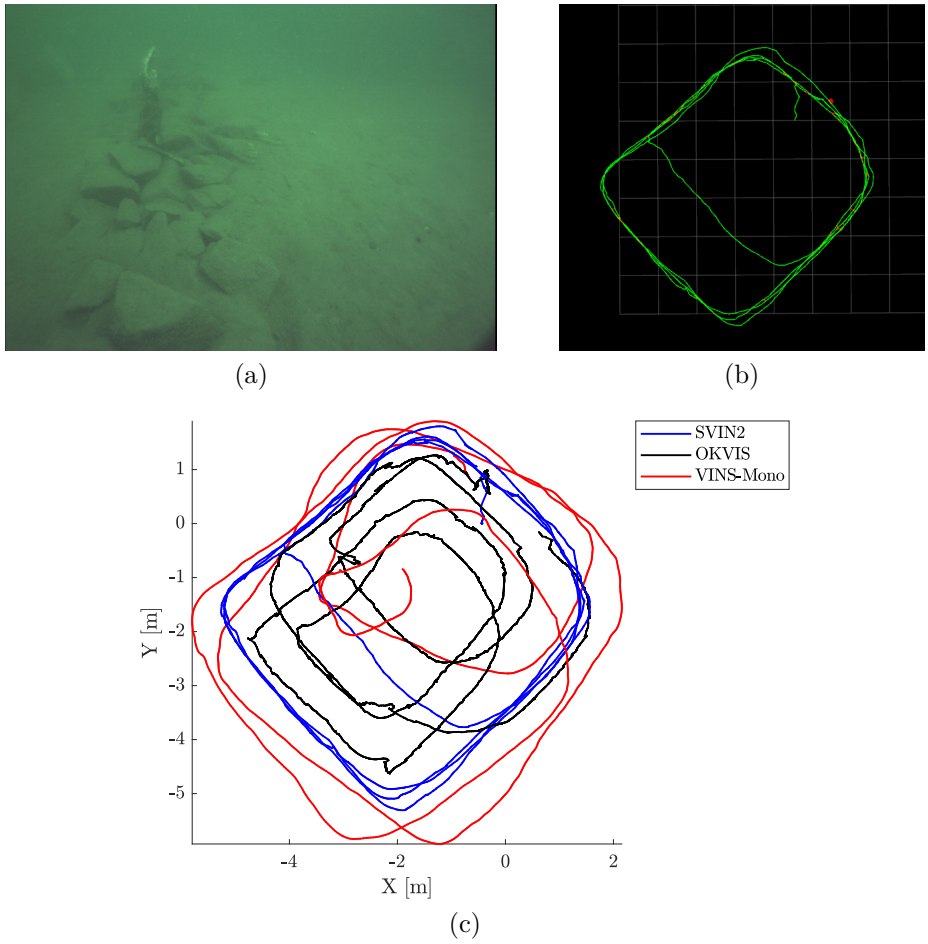


Figure 4.10: (a) Aqua2 in a fake cemetery, Lake Jocassee, SC, USA; trajectories from SVIn2 with visual, inertial, and depth sensor (no sonar data has been used) shown in rviz (b) and aligned trajectories from SVIn2 with Sonar and depth disabled, OKVIS, and VINS-Mono (c) are displayed.

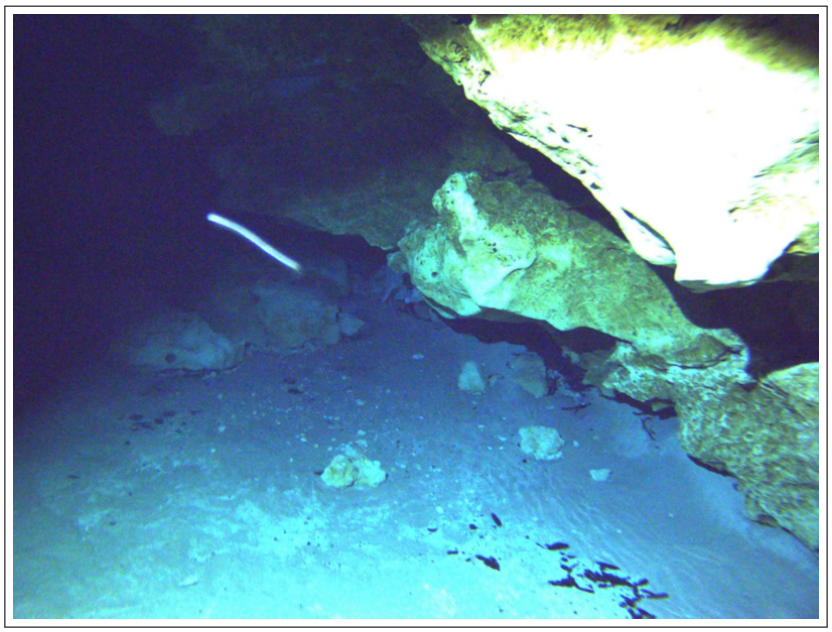
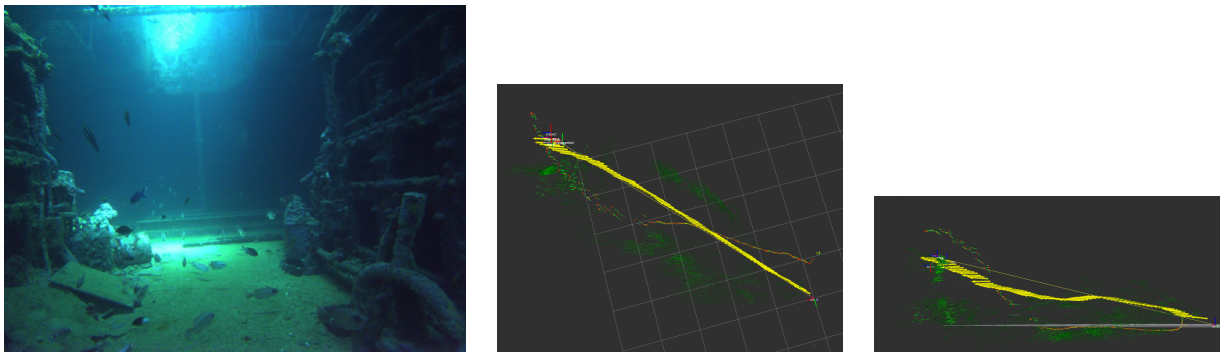
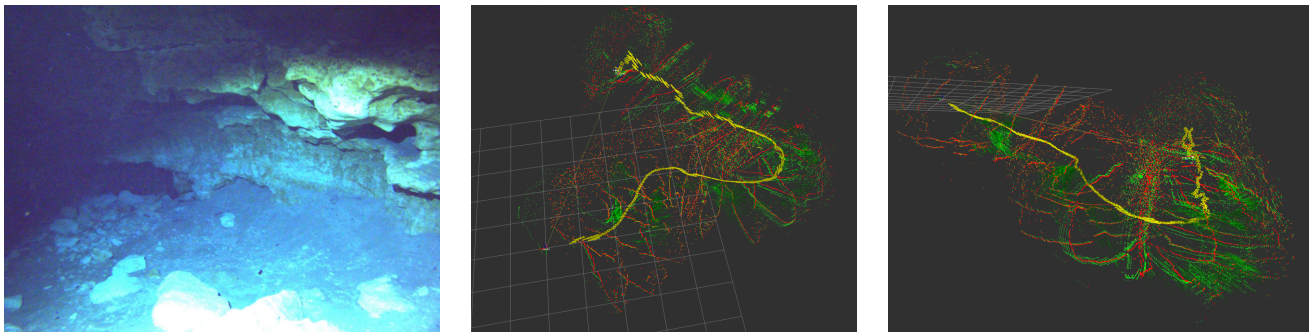


Figure 4.11: A small particle reflecting back at high speed generating a blurry streak. In addition light reflecting back from a nearby surface completely saturates the camera.



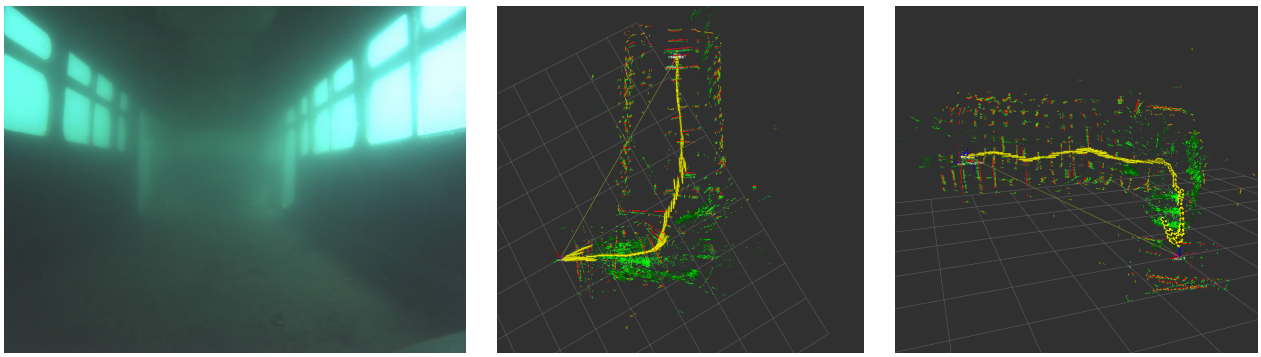
(a) (b) (c)

Figure 4.12: Bajan Queen artificial reef (shipwreck) in Carlisle Bay, Barbados. (a) Sample image of the data collected inside the wreck (beginning of trajectory). (b) Top view of the reconstruction.



(a) (b) (c)

Figure 4.13: Underwater cave, Ballroom Ginnie cavern at High Springs, FL, USA. (a) Sample image of the data collected inside the cavern. (b) Top view of the reconstruction. (c) Side view of the reconstruction.



(a) (b) (c)

Figure 4.14: Sunken bus, Fantasy Lake Scuba Park, NC, USA. (a) Sample image of the data collected from inside the bus. (b) Top view of the reconstruction. (c) Side view of the reconstruction, note the stairs detected by visual features at the right side of the image.

Chapter 5

Contour based Reconstruction of Underwater Structures Using Sonar, Visual, Inertial, and Depth Sensor

5.1 Introduction

Underwater cave exploration is one of the most extreme adventures pursued by humans in [Exley 1977]. It is a dangerous activity with more than 600 fatalities, since the beginning of underwater cave exploration, that currently attracts many divers. Generating models of the connectivity between different underwater cave systems together with data on the depth, distribution, and size of the underwater chambers is extremely important for fresh water managements in [*Climate Change and Sea-Level Rise in Florida: An Update of "The Effects of Climate Change on Florida's Ocean and Coastal Resources."* 2010], environmental protection, and resource utilization in [Xu et al. 2016]. In addition, caves provide valuable historical evidence as they present an undisturbed time capsule in [Abbott 2014], and information about geological processes in [Kresic and Mikszewski 2013].

Before venturing beyond the light zone with autonomous robots, it is crucial to ensure that localization and mapping abilities have been developed and are adequately robust. Constructing a map of an underwater cave presents many challenges. First of all, vision underwater is plagued by limited visibility, color absorption, hazing, and lighting variations. Furthermore, the absence of natural light inside underwater caves makes localization and mapping more difficult; however, the use of an artificial light can be used to infer the structure in [Weidner et al. 2017]. The most common underwater mapping sensor is based on sonar, which, when mounted on a moving platform, requires a secondary sensor to provide a common frame of reference for the range measurements collected over time. Furthermore, the majority of sonar sensors generate multiple returns in enclosed spaces making mapping caves extremely difficult.

In our earlier work, the cone of light perceived by a stereo camera was used to reconstruct offline the boundaries of a cave in Mexico in [Weidner et al. 2017]. No other sensor was available and the stereo-baseline of 0.03 m limited the accuracy of the

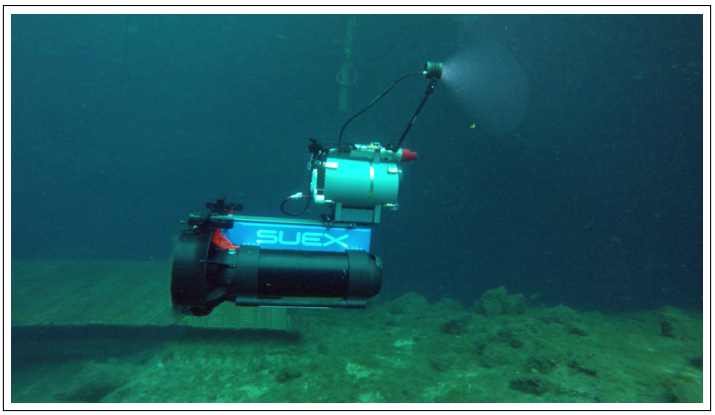


Figure 5.1: The stereo, inertial, depth, and acoustic sensor suite mounted on a dual diver propulsion vehicle (DPV) equipped with a flashlight, in front of the Blue Grotto cavern.

reconstruction for objects further than a couple of meters. More recently, augmenting the visual-inertial state estimation package OKVIS in [Leutenegger et al. 2015], we fused visual and inertial data together with acoustic range measurements from a pencil beam sonar, which provide more reliable distance estimate of features. This allows a more robust and reliable state estimation in [Rahman, Quattrini Li, and Rekleitis 2018b]. One of the limitations is the granularity of the resulting 3D point cloud: only few keypoints are typically tracked, resulting in very sparse 3D point cloud, which cannot be directly used, for example, by an Autonomous Underwater Vehicle (AUV) to navigate and avoid obstacles. Applying a direct-based method, such as LSD-SLAM in [Engel, Schöps, and Cremers 2014], is not straightforward, given the sharp changes in illumination in the underwater scene. A fundamental difference with most vision based estimation approaches is that in a cave environment, the light source is constantly moving thus generating shadows that also move. Consequently the majority of the strong features cannot be used for estimating the pose of the camera.

In this paper, we propose a novel system that is able to track the state estimate and at the same time improve the 3-D reconstruction from visual edge based information in the cave boundaries.

In particular, the proposed approach for real-time reconstruction of the cave environment with medium density is based on an underwater SLAM system that combines acoustic (sonar range), visual (stereo camera), inertial (linear accelerations and angular velocities), and depth data to estimate the trajectory of the employed sensor suite. The inspiration for a denser point cloud comes from the following observation: visual features on the boundaries created by shadows, occlusion edges, and the boundaries of the artificial illumination (video light) – see Fig. 5.1 – are all located at the floor, ceiling, and walls of the cave. The point cloud resulting from such edges is then optimized in a local bundle adjustment, and can be used for providing a denser reconstruction, enabling the deployment of AUVs like Aqua2 in [Dudek et al. 2005] with advanced swimming gaits in [Meger et al. 2015], navigating around obstacles without disturbing the sediment at the bottom. Experiments in caverns and caves validate the proposed approach.

The paper is structured as follows. In the next section, we present related work, specifically focusing on state estimation and 3D reconstruction. Section 5.2 describes the proposed method. Experimental results are presented in Section 5.3. Section 5.4 concludes the paper.

5.2 Technical Approach

The proposed approach augments in [Rahman, Quattrini Li, and Rekleitis 2018b; Rahman, Quattrini Li, and Rekleitis 2018c] to generate real-time a denser reconstruction of underwater structures exploiting the boundaries of the structure and the cone-of-light. For completeness, we briefly introduce the system hardware and visual inertial method that includes acoustic and depth measurements. Then, we describe the proposed 3D reconstruction based on contour matching and the local optimization of such point cloud.

5.2.1 System Overview

The target hardware system is composed of a stereo camera, mechanical scanning profiling Sonar, IMU, pressure sensor, and an on-board computer. This is part for example of a custom-made sensor suite – which can be carried by divers as well as can be mounted on a single or dual Diver Propulsion Vehicle (DPV) in [Rahman, Quattrini Li, and Rekleitis 2018a] – or an AQUA2 AUV in [Dudek et al. 2005], which have been used for underwater reconstruction. The hardware was designed with cave mapping as the target application. As such, the sonar scanning plane is parallel to the image plane which provides data at a maximum of 6 m range, scanning in a plane over 360°, with angular resolution of 0.9°.

5.2.2 Notations and States

The reference frames associated to each sensor and the world are denoted as C for Camera, I for IMU, S for Sonar, D for Depth, and W for World. Let us denote ${}_X\mathbf{T}_Y = [{}_X\mathbf{R}_Y|{}_X\mathbf{p}_Y]$ the homogeneous transformation matrix between two arbitrary coordinate frames X and Y, where ${}_X\mathbf{R}_Y$ represents the rotation matrix with corresponding quaternion ${}_X\mathbf{q}_Y$ and ${}_X\mathbf{p}_Y$ denotes the position vector.

The state of the robot R is denoted as $\mathbf{x}_R$:

$$\mathbf{x}_R = [{}_W\mathbf{p}_I^T, {}_W\mathbf{q}_I^T, {}_W\mathbf{v}_I^T, {\mathbf{b}_g}^T, {\mathbf{b}_a}^T]^T \tag{5.1}$$

It contains the position ${}_W\mathbf{p}_I$, the quaternion ${}_W\mathbf{q}_I$, the linear velocity ${}_W\mathbf{v}_I$. All of them are in the IMU reference frame I with respect to the world reference frame W. In addition, the gyroscopes and accelerometers bias $\mathbf{b}_g$ and $\mathbf{b}_a$ are also estimated and stored in the state vector.

The corresponding error-state vector is defined in minimal coordinates, while the perturbation takes place in the tangent space:

$$\delta\boldsymbol{\chi}_R = [\delta\mathbf{p}^T, \delta\mathbf{q}^T, \delta\mathbf{v}^T, \delta{\mathbf{b}_g}^T, \delta{\mathbf{b}_a}^T]^T \tag{5.2}$$

5.2.3 Tightly-coupled Non-Linear Optimization Problem

The cost function $J(\mathbf{x})$ for the tightly-coupled non-linear optimization includes the reprojection error $\mathbf{e}_r$, the IMU error $\mathbf{e}_s$, sonar error $\mathbf{e}_t$, and the depth error e_u:

$$\begin{aligned} J(\mathbf{x}) &= \sum_{i=1}^{2}\sum_{k=1}^{K}\sum_{j\in\mathcal{J}(i,k)} \mathbf{e}_r^{i,j,k^T}\mathbf{P}_r^k\mathbf{e}_r^{i,j,k} + \sum_{k=1}^{K-1}\mathbf{e}_s^{k^T}\mathbf{P}_s^k\mathbf{e}_s^k \\ &+ \sum_{k=1}^{K-1}\mathbf{e}_t^{k^T}\mathbf{P}_t^k\mathbf{e}_t^k + \sum_{k=1}^{K-1} e_u^{k^T}P_u^k e_u^k \end{aligned} \tag{5.3}$$

with i denoting the camera index – $i = 1$ for left, $i = 2$ for right camera in a stereo camera – and landmark index j observed in the k^{th} camera frame. $\mathbf{P}_r^k$, $\mathbf{P}_s^k$, $\mathbf{P}_t^k$, and P_u^k denote the information matrix of visual landmarks, IMU, sonar range, and depth measurement for the k^{th} frame respectively.

The reprojection error describes the difference between a keypoint measurement in camera coordinate frame C and the corresponding landmark projection according to the stereo projection model. The IMU error term combines all accelerometer and gyroscope measurements by *IMU pre-integration* in [Forster et al. 2017a] between successive camera measurements and represents the *pose*, *speed and bias* error between the prediction based on previous and current states. Both reprojection error and IMU error term follow the formulation by Leutenegger et. al.in [Leutenegger et al. 2015].

The sonar range error, introduced in the previous chapter [Rahman, Quattrini Li, and Rekleitis 2018b], represents the difference between the 3D point that can be derived from the range measurement and a corresponding visual feature in 3D.

The depth error term can be calculated as the difference between the rig position along the z direction and the water depth measurement provided by a pressure sensor. Depth values are extracted along the *gravity* direction which is aligned with the z of the world W – observable due to the tightly coupled IMU integration. This can correct the position of the robot along the z axis.

Ceres Solver nonlinear optimization framework Agarwal, Mierle, et al. 2015 optimizes $J(\mathbf{x})$ then to estimate the state of the system.

5.2.4 Feature Selection and 3D Reconstruction from Stereo Contour Matching

To ensure that the VIO system and the 3D reconstruction can be run in real-time in parallel, we replaced the OKVIS feature detection method with the one described in [Shi et al. 1994], which provides a short list of the most prominent features based on the *corner response* function in the images. This reduces the computation in the *frontend* tracking and, as shown in the results, retains the same accuracy with less computational requirements.

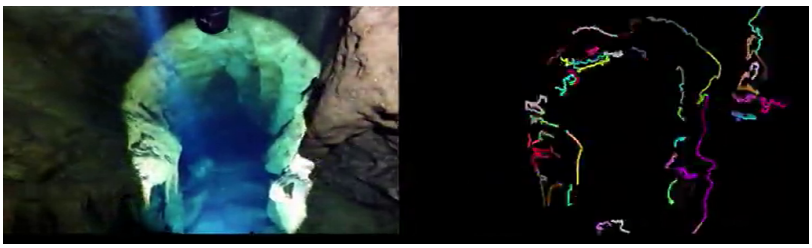


Figure 5.2: Image in a cave and the detected contours.

We included a real-time stereo contour matching algorithm followed by an outlier rejection mechanism to produce the point-cloud on the contour created by the light; see Fig. 5.4c for an example of all the edge-features detected. The approach of Weidner et. al in [Weidner et al. 2017] has been adapted for the contours from the intersection of the cone of light with the cave wall; see Fig. 5.2 for the extracted contours from an underwater cave. In particular, adaptive thresholding the images based on the light and dark areas ensures that the illuminated areas are clearly defined. In our current work, we also found that sampling from pixels which have rich gradient, e.g., edges provides better and denser point-cloud reconstructions. As such, both types of edges – one marking the boundaries between the light and dark areas and the other from visible cave walls – are used to reconstruct the 3-D map of the cave. The

overview of the augmenting Stereo Contour Matching method in our tightly-coupled Sonar-Visual-Inertial-Depth optimization framework is as follows.

For every *frame* in the local optimization window, a noisy edge map is created from the edges described above, followed by a filtering process to discard short contours by calculating their corresponding bounding boxes and only keeping the largest third percentile. This method retains the highly defined continuous contours of the surroundings while eliminating spurious false edges, thus allowing to use the pixels on them as good features to be used in the reconstruction. In a stereo frame, for every image point on the contour of the left image a BRISK feature descriptor is calculated and matched against the right image searching along the epipolar line. Then a sub-pixel accurate localization of the matching disparity is performed. Another layer of filtering is done based on the grouping of the edge detector, i.e., keeping only the consecutive points belonging to the same contour in a stereo pair. These stereo contour matched features along with depth estimation is projected into 3-D which are projected back for checking the reprojection error consistency resulting into a point-cloud with very low reprojection error.

The reason behind choosing stereo matched contour features rather than tracking them using a semi-direct method is to avoid any spurious edge detection due to lighting variation in consecutive images, which could lead to erroneous estimation or even tracking failure. The performance of SVO in [Forster et al. 2017b], an open-source state-of-the-art semi-direct method, in underwater datasets in [Quattrini Li et al. 2016; Joshi et al. 2019] validates this statement. In addition, though indirect feature extractors and descriptors are invariant to photometric variations to some extent, using a large number of features for tracking and thus using them for reconstruction is unrealistic due to the computational complexity of maintaining them.

5.2.5 Local Bundle Adjustment (BA) for Contour Features

In the current optimization window, a local BA is performed for all newly detected stereo contour matched features and the keyframes they are observed in, to achieve an optimal reconstruction. A joint non-linear optimization is performed for refining k^{th} keyframe pose ${}_W\mathbf{T}_{C_i}{}^k$ and homogeneous *landmark* j in world coordinate W, ${}_W\boldsymbol{l}^j = [{l_x}^j, {l_y}^j, {l_z}^j, {l_w}^j]$ minimizing the cost function:

$$J(\mathbf{x}) \quad = \quad \sum_{j,k} \rho(\mathbf{e}^{j,k^T} \mathbf{P}^{j,k} \mathbf{e}^{j,k}) \tag{5.4}$$

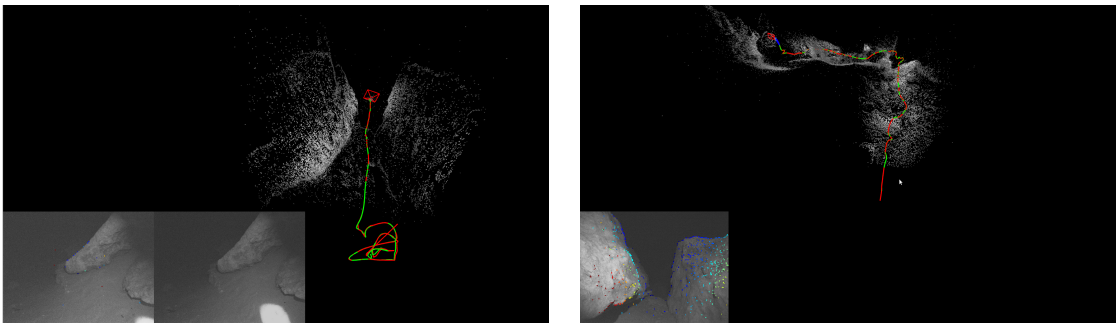
Hereby $\mathbf{P}^{j,k}$ denotes the information matrix of associated landmark measurement, ρ is the Huber loss function to *down-weight* outliers. The reprojection error, $\mathbf{e}^{j,k}$ for landmark j with matched keypoint measurement $\mathbf{z}_{j,k}$ in image coordinate in the respective camera i is defined as:

$$\mathbf{e}^{j,k} \quad = \quad \mathbf{z}^{j,k} - \mathbf{h}_i({}_W\mathbf{T}_{C_i}{}^k, {}_W\boldsymbol{l}^j) \tag{5.5}$$

with camera projection model $\mathbf{h}_i$. We used Levenberg-Marquardt to solve local BA problem which obtains a good estimation for the non-linear optimization system.

5.3 Experimental Result

The experimental data were collected using a custom made sensor suite in [Rahman, Quattrini Li, and Rekleitis 2018a] consisting of a stereo camera, an IMU, a depth sensor and a mechanical scanning Sonar, as described in Section 5.2.1. More specifically, two USB-3 uEye cameras in a stereo configuration provide data at 15 Hz, an IMAGENEX 831L mechanical scanning Sonar sensor acquires a full 360° scan every four seconds; the Bluerobotics Bar30 pressure sensor provides depth data at 1 Hz; a MicroStrain 3DM-GX4-15 IMU generates inertial data at 100 Hz; and an Intel NUC running Linux and ROS consolidates all the data. A video light is attached to the unit to provide artificial illumination of the scene. The Sonar is mounted on top of



(a) (b)

Figure 5.3: Partial trajectories generated by DSO. (a) Incorrect odometry and failing to track just after a few seconds and (b) longer trajectory after starting at a place with better illumination which also fails later on.

the main unit which contains the remaining electronics. In Fig. 5.6(a,b) the unit can be seen deployed in two different modes: hand-held by a diver and mounted on a Diver Propulsion Vehicle (DPV).

In the following, we present, first, preliminary experiments with DSO in [Engel, Koltun, and Cremers 2018] showing the problem with photometric consistency, and, second, a qualitative result of the proposed approach in different underwater environments.

5.3.1 Comparison with DSO

Fig. 5.3 shows the result of DSO in the underwater cave dataset in two different runs, Fig. 5.3a and Fig. 5.3b. DSO did not track for the full length of cave, instead it was able to keep track just for a small segment due to the variation of the light and hence violating the *photometric consistency* assumption of a direct method. Also, the *initialization* method is critical as it requires mainly translational movement and a very small rotational change due to the fact that it is a pure monocular visual SLAM. We ran DSO with different starting point of the dataset to have a better initialization, the best one we got in Fig. 5.3b – eventually failed too because of the poor lighting.

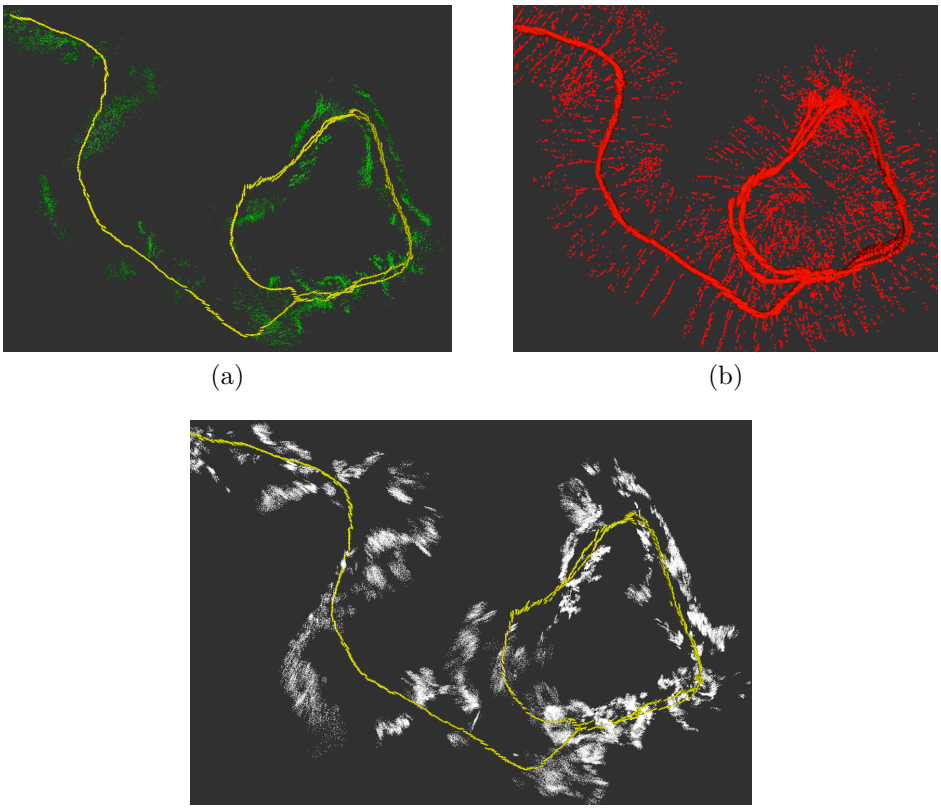


Figure 5.4: (a) Odometry using only a few strong features (green) for tracking. (b) Scanning Sonar measurements (red) aligned along the trajectory. (c) Reconstruction of the cave using the edges detected in the stereo contour points (gray).

5.3.2 Odometry and 3D Cave-Wall Reconstruction

The ballroom at Ginnie Springs, FL, is a cavern open to divers with no cave-diving training. It provides a safe locale to collect data in an underwater cave environment. From entering the cavern at a depth of seven meters, the sensor was taken down to fifteen meters, and then a closed loop trajectory was traversed three times. As there is no ground truth available underwater, such as a motion capture system, we validate our approach from the information collected by the divers during the data collection procedure. The length of the trajectory produced by our method is 87 meters, consistent with the measure from the divers.

Fig. 5.4 shows the whole trajectory with the different point clouds generated by

the features used for tracking, Sonar data, and stereo contour matching. Keeping a small set of features for only tracking helps to run the whole system in real-time. As shown in the figure, Sonar provides a set of sparse but robust points using *range* and *head_position* information. Finally, the stereo contour matched point generates a denser point-cloud to represent the cave environment. Fig. 5.5 highlights some specific sections of the cavern, with the image and the corresponding reconstruction – in gray, the points from the contours; in red the points from the Sonar. As it can be observed, our proposed method enhances the reconstruction with a dense point cloud; for example rocks and valleys are clearly visible in Fig. 5.5.

5.4 Discussion

The proposed system improves the point cloud reconstruction and is able to perform in real time even with additional capabilities. One of the lessons learned during experimental activities is that the position of the light affects also the quality of the reconstruction. In the next version of the sensor suite, we plan to mount the dive light in a fixed position so that the cone of light can be predicted according to the characteristics of the dive light. Furthermore, setting the maximum distance of the Sonar according to the specific environment improves the range measurements obtainable.

We are currently deploying the sensor suite either hand-held by a diver – see Fig. 5.6a – or mounted on a DPV – see 5.6b – in a variety of locations. Future plans are to deploy the sensor suite on a dual DPV which will provide greater stability – see Fig. 5.1 for preliminary tests. Furthermore, the sonar can be deployed on an AUV, such as an Aqua2 in [Dudek et al. 2005] vehicle – see 5.6c, for autonomous operations. It is worth noting that the sensor suite utilizes the same hardware with an Aqua2 AUV for maximum compatibility.

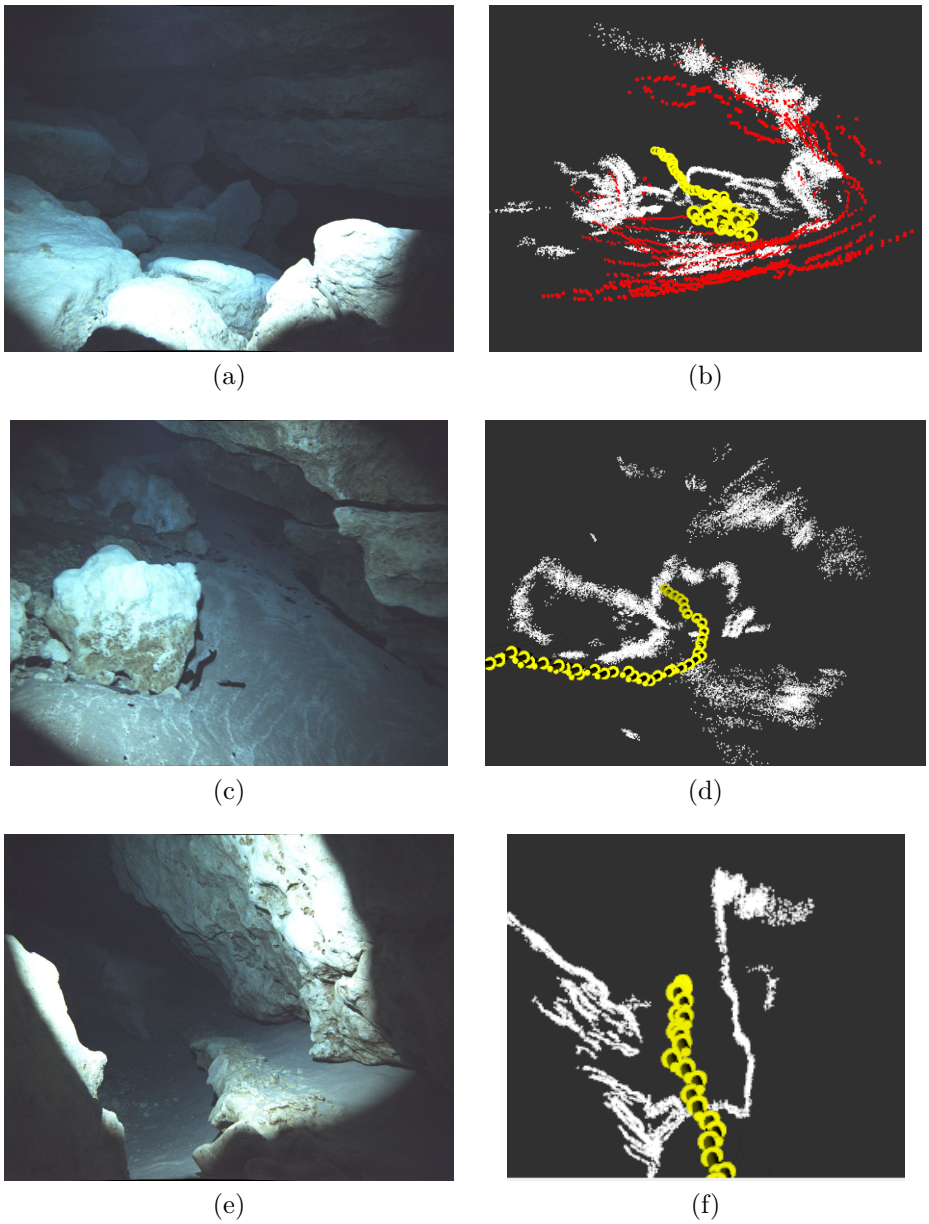
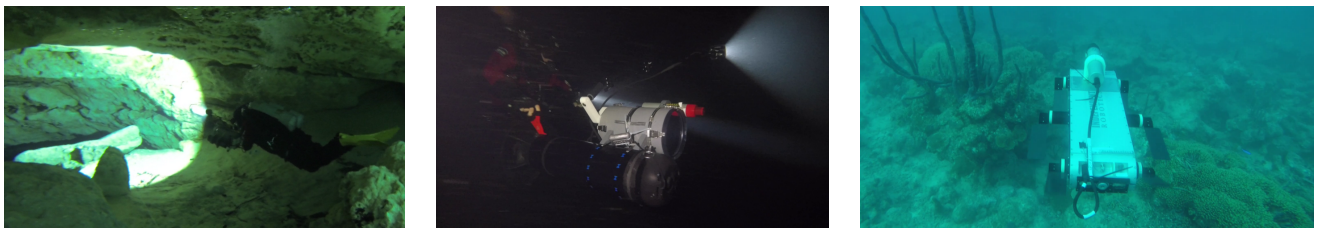


Figure 5.5: Stereo contour reconstruction results in (b), (d), (f) and the corresponding images in (a), (c), (e) respectively.



(a) (b) (c)

Figure 5.6: Data collection approaches: (a) Diver holds the sensor swimming through the cave. (b) Sensor suite mounted on a DPV. (c) an Aqua 2 vehicle in [Dudek et al. 2005] with similar hardware carrying the scanning sonar collects data over a coral reef.

Chapter 6

Conclusions

As vision based stated estimation achieves a certain degree of maturity, more sensors are being integrated. Extending the well studied problem of Visual Inertial integration, we introduce a new sensor, a mechanical scanning sonar, which returns range measurements based on acoustic information. While the primary motivation of our work has been the mapping of underwater caves in [Weidner et al. 2017], the technique was tested in different environments, including the a shipwreck at the clear waters of Barbados, to artificial wrecks in the lakes of the Carolinas. A novel approach of merging sonar points with visual features is used to extend the pose graph generated for applying a global nonlinear optimization. The integration of the range data in the popular optimizer of Ceres in [Agarwal, Mierle, et al. 2015] resulted in scale estimation improvements.

During the different experiments, it became clear that a minimum visibility and clarity in the visual data is required for basic performance, however, the data used degraded to a degree not often seen in VO or VIO approaches. Moreover, the use of a strong video light while necessary in the cave environment, it requires careful calibration of its position in order to not saturate the camera. Furthermore, different surfaces resulted in different reflectance properties of the acoustic signal; we are currently analyzing the sonar data to improve the quality.

Integration of multiple sensors will improve the quality of the estimation in addition to the density of the reconstruction. A variety of domains will be affected with underwater archaeology and speleology being the primary areas. The resulting tech-

nology will be integrated to existing AUVs and ROVs for improving their autonomous capabilities.

Bibliography

Abbott, Alison (May 2014). "Mexican skeleton gives clue to American ancestry". In: *Nature News*. Springer Nature.

Agarwal, Sameer, Keir Mierle, et al. (2015). *Ceres Solver*. `http://ceres-solver.org`.

Alvarez, A et al. (2005). "Folaga: a very low cost autonomous underwater vehicle for coastal oceanography". In: *International Federation of Automatic Control Congress (IFAC)*, pp. 31–36.

Anqi Xu and contributors (2018). *ueye_cam package*. `https://github.com/anqixu/ueye_cam`. Accessed: 2018-08-11.

Autonomous Field Robotics Lab (2018). *Stereo Rig Sensor Documentation*. `https://afrl.cse.sc.edu/afrl/resources/StereoRigWiki/`. Accessed: 2018-08-14.

Badino, Hernán, Akihiro Yamamoto, and Takeo Kanade (2013). "Visual odometry by multi-frame feature integration". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 222–229.

Beall, Chris et al. (2011). "Bundle adjustment in large-scale 3D reconstructions based on underwater robotic surveys". In: *MTS/IEEE OCEANS, Spain*, pp. 1–6.

Bellavia, Fabio, Marco Fanfani, and Carlo Colombo (2015). "Selective visual odometry for accurate AUV localization". In: *Autonomous Robots*, pp. 1–11.

Bloesch, Michael et al. (2017). "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback". In: *Int. J. Robot. Res.* 36, pp. 1053–1072. DOI: `10.1177/0278364917728574`.

Burri, Michael et al. (2016). "The EuRoC micro aerial vehicle datasets". In: *Int. J. Robot. Res.* 35.10, pp. 1157–1163. DOI: `10.1177/0278364915620033`.

C. McKinlay (Apr. 2015). *Woodville Karst Plain Project (WKPP)*. URL:`http://www.wkpp.org`.

Chen, Shi-Feng and Jun-Zhi Yu (2014). "Underwater cave search and entry using a robotic fish with embedded vision". In: *Chinese Control Conference (CCC)*, pp. 8335–8340.

Climate Change and Sea-Level Rise in Florida: An Update of "The Effects of Climate Change on Florida's Ocean and Coastal Resources." (2010). Tech. rep. Tallahasee, FL: Florida Ocean and Coastal Council.

Corke, Peter et al. (2007). "Experiments with underwater robot localization and tracking". In: *Proc. ICRA*. IEEE, pp. 4556–4561.

Cummins, Mark and Paul Newman (2008). "FAB-MAP: Probabilistic localization and mapping in the space of appearance". In: *Int. J. Robot. Res.* 27.6, pp. 647–665.

— (2011). "Appearance-only SLAM at large scale with FAB-MAP 2.0". In: *Int. J. Robot. Res.* 30.9, pp. 1100–1123.

Davison, Andrew J et al. (2007). "MonoSLAM: Real-time single camera SLAM". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 6, pp. 1052–1067.

Delmerico, J. and D. Scaramuzza (2018). "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots". In: *Proc. ICRA*.

Dudek, Gregory et al. (2005). "A Visually Guided Swimming Robot". In: *Proc. IROS*, pp. 1749–1754.

Engel, Jakob, Vladlen Koltun, and Daniel Cremers (2018). "Direct sparse odometry". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.3, pp. 611–625.

Engel, Jakob, Thomas Schöps, and Daniel Cremers (2014). "LSD-SLAM: Large-scale direct monocular SLAM". In: *European conference on computer vision*. Springer, pp. 834–849.

Exley, Sheck (1977). *Basic Cave Diving: A Blueprint for Survival*. ISBN 99946-633-7-2. National Speleological Society Cave Diving Section.

Fallon, Maurice F et al. (2013). "Relocating underwater features autonomously using sonar-based SLAM". In: *IEEE Journal of Oceanic Engineering* 38.3, pp. 500–513.

Fiala, Mark (2004). "Artag revision 1, a fiducial marker system using digital techniques". In: *National Research Council Publication* 47419, pp. 1–47.

Folkesson, John et al. (2007). "Feature tracking for underwater navigation using sonar". In: *Proc. IROS*. IEEE, pp. 3678–3684.

Ford, D. C. and P. W. Williams (1994). *Karst Geomorphology and Hydrology*. Chapman & Hall.

Forster, Christian et al. (2017a). "On-Manifold Preintegration for Real-Time Visual–Inertial Odometry". In: *IEEE Trans. Robot.* 33.1, pp. 1–21.

Forster, Christian et al. (2017b). "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems". In: *IEEE Trans. Robot.* 33.2. DOI: 10.1109/TRO.2016.2623335.

Gálvez-López, Dorian and Juan D Tardos (2012). "Bags of binary words for fast place recognition in image sequences". In: *IEEE Trans. Robot.* 28.5, pp. 1188–1197.

Gary, Marcus et al. (2008). "3D mapping and characterization of sistema Zacatón from DEPTHX (DEep Phreatic THermal eXplorer)". In: *Proc. of KARST: Sinkhole Conference ASCE*.

Geiger, Andreas et al. (2013). "Vision meets Robotics: The KITTI Dataset". In: *Int. J. Robot. Res.* 32.11, pp. 1231–1237.

Giguere, Philippe et al. (2009). "Unsupervised learning of terrain appearance for automated coral reef exploration". In: *Proc. CRV*. IEEE, pp. 268–275.

Gulden, B. (Apr. 2015). *WORLD LONGEST UNDERWATER CAVES*. URL:http://www.caverbob.com/uwcaves.htm.

Henderson, Jon et al. (2013). "Mapping submerged archaeological sites using stereo-vision photogrammetry". In: *International Journal of Nautical Archaeology* 42.2, pp. 243–256.

Hesch, Joel A et al. (2012). "Observability-constrained vision-aided inertial navigation". In: *University of Minnesota, Dept. of Comp. Sci. & Eng., MARS Lab, Tech. Rep* 1, p. 6.

Hildebrandt, Marc and Frank Kirchner (2010). "Imu-aided stereo visual odometry for ground-tracking auv applications". In: *OCEANS 2010 IEEE-Sydney*. IEEE, pp. 1–8.

Hogue, Andrew, Andrew German, and Michael Jenkin (2007). "Underwater environment reconstruction using stereo and inertial data". In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, pp. 2372–2377.

Howard, Andrew (2008). "Real-time stereo visual odometry for autonomous ground vehicles". In: *Proc. IROS*. IEEE, pp. 3946–3952.

Johannsson, Hordur et al. (2010). "Imaging sonar-aided navigation for autonomous underwater harbor surveillance". In: *Proc. IROS*. IEEE, pp. 4396–4403.

Johnson-Roberson, Matthew et al. (2010). "Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys". In: *J. Field Robot.* 27.1, pp. 21–51.

Jones, Eagle S and Stefano Soatto (2011). "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach". In: *Int. J. Robot. Res.* 30.4, pp. 407–430.

Joshi, Bharat et al. (2019). "Experimental Comparison of open source Vision-Inertial-Based State Estimation Algorithms". In: *Proc. IROS*. (under review).

Kelly, Jonathan and Gaurav S Sukhatme (2011). "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration". In: *Int. J. Robot. Res.* 30.1, pp. 56–79. DOI: 10.1177/0278364910382802.

Kitt, Bernd, Andreas Geiger, and Henning Lategahn (2010). "Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme." In: *Intelligent Vehicles Symposium*, pp. 486–492.

Klein, G. and D. Murray (2007). "Parallel Tracking and Mapping for Small AR Workspaces". In: *IEEE and ACM Int. Symp. on Mixed and Augmented Reality*, pp. 225–234. DOI: 10.1109/ISMAR.2007.4538852.

Konolige, Kurt, Motilal Agrawal, and Joan Sola (2010). "Large-scale visual odometry for rough terrain". In: *Robotics research*. Springer, pp. 201–212.

Kresic, N. and A. Mikszewski (2013). *Hydrogeological Conceptual Site Models: Data Analysis and Visualization*. Boca Raton, LA: CRC Press.

Kumar Robotics (2018). *imu_3dm_gx4 package*. https://github.com/KumarRobotics/imu_3dm_gx4. Accessed: 2018-08-11.

Lane, Ed (2001). *The Spring Creek Submarine Springs Group, Wakulla County, Florida*. Tech. rep. Special Publication 47. Tallahasee, Fl: Florida Geological Survey.

Lee, Chong-Moo et al. (2005). "Underwater navigation system based on inertial sensor and doppler velocity log using indirect feedback Kalman filter". In: *International Journal of Offshore and Polar Engineering* 15.02.

Leedekerken, Jacques C, Maurice F Fallon, and John J Leonard (2014). "Mapping complex marine environments with autonomous surface craft". In: *Proc. ISER*, pp. 525–539.

Leonard, John J and Hugh F Durrant-Whyte (2012). *Directed sonar sensing for mobile robot navigation*. Vol. 175. Springer Science & Business Media.

Leutenegger, Stefan et al. (2015). "Keyframe-based visual-inertial odometry using nonlinear optimization". In: *Int. J. Robot. Res.* 34.3, pp. 314–334.

Mallios, Angelos et al. (2016). "Toward autonomous exploration in confined underwater environments". In: *J. Field Robot.* 33.7, pp. 994–1012.

Mallios, Angelos et al. (2017). "Underwater caves sonar data set". In: *Int. J. Robot. Res.* 36.12, pp. 1247–1251.

Meger, David et al. (2015). "Learning legged swimming gaits from experience". In: *Proc. ICRA*, pp. 2332–2338.

Merrell, Paul et al. (2007). "Real-time visibility-based fusion of depth maps". In: *Proc. ICCV*. IEEE, pp. 1–8.

Mourikis, Anastasios I and Stergios I Roumeliotis (2007). "A multi-state constraint Kalman filter for vision-aided inertial navigation". In: *Proc. ICRA*. IEEE, pp. 3565–3572.

Mur-Artal, Raúl, J. M. M. Montiel, and Juan D. Tardós (2015a). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robot.* 31.5, pp. 1147–1163.

— (2015b). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robot.* 31.5, pp. 1147–1163.

Mur-Artal, Raúl and Juan D Tardós (2017). "Visual-inertial monocular SLAM with map reuse". In: *IEEE Robot. Autom. Lett.* 2.2, pp. 796–803.

Oliver, Kenton, Weilin Hou, and Song Wang (2010). "Image feature detection and matching in underwater conditions". In: *Ocean Sensing and Monitoring II*. Vol. 7678. International Society for Optics and Photonics, 76780N.

Oskiper, Taragay et al. (2007). "Visual odometry system using multiple stereo cameras and inertial measurement unit". In: *Proc. CVPR*. IEEE, pp. 1–8.

Qin, Tong, Peiliang Li, and Shaojie Shen (2018). "VINS-Mono: A robust and versatile monocular visual-inertial state estimator". In: *IEEE Trans. Robot.* 34.4, pp. 1004–1020.

Quattrini Li, Alberto et al. (2016). "Experimental Comparison of open source Vision based State Estimation Algorithms". In: *Proc. ISER*.

Rahman, Sharmin, Alberto Quattrini Li, and Ioannis Rekleitis (2018a). "A Modular Sensor Suite for Underwater Reconstruction". In: *MTS/IEEE Oceans Charleston*, pp. 1–6.

— (2018b). "Sonar Visual Inertial SLAM of Underwater Structures". In: *Proc. ICRA*.

— (2018c). "SVIn2: Sonar Visual-Inertial SLAM with Loop Closure for Underwater Navigation". In: *CoRR* abs/1810.03200. arXiv: 1810.03200. URL: http://arxiv.org/abs/1810.03200.

Research group of Prof. Kostas Daniilidis (2018). *Monocular MSCKF ROS node*. https://github.com/daniilidis-group/msckf_mono.

Richmond, Kristof et al. (2018). "SUNFISH®: A human-portable exploration AUV for complex 3D environments". In: *MTS/IEEE OCEANS Charleston*, pp. 1–9.

Rigby, Paul, Oscar Pizarro, and Stefan B Williams (2006). "Towards geo-referenced AUV navigation through fusion of USBL and DVL measurements". In: *OCEANS*, pp. 1–6.

Roman, Chris et al. (2000). "A new autonomous underwater vehicle for imaging research". In: *MTS/IEEE OCEANS Conference and Exhibition*. Vol. 1, pp. 153–156.

Sáez, Juan Manuel et al. (2006). "Underwater 3D SLAM through entropy minimization". In: *Proc. ICRA*. IEEE, pp. 3562–3567.

Salvi, Joaquim et al. (2008). "Visual SLAM for underwater vehicles using video velocity log and natural landmarks". In: *MTS/IEEE OCEANS*, pp. 1–6.

Sattar, Junaed et al. (2007). "Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction". In: *Proc. CRV*, pp. 165–174.

Sattar, Junaed et al. (2008). "Enabling Autonomous Capabilities in Underwater Robotics". In: *Proc. IROS*, pp. 3628–3634.

Schonberger, Johannes L and Jan-Michael Frahm (2016). "Structure-from-motion revisited". In: *Proc. CVPR*, pp. 4104–4113.

Shi, Jianbo et al. (1994). “Good features to track”. In: *Proc. CVPR*. IEEE, pp. 593–600.

Shkurti, Florian et al. (2011). “State estimation of an underwater robot using visual and inertial information”. In: *Proc. IROS*, pp. 5054–5060.

Skaff, S., J.J. Clark, and Ioannis Rekleitis (Sept. 2008). “Estimating Surface Reflectance Spectra for Underwater Color Vision”. In: *British Machine Vision Conference (BMVC)*. Leeds, U.K., pp. 1015–1024.

Snavely, Noah, Steven M Seitz, and Richard Szeliski (2006). “Photo tourism: exploring photo collections in 3D”. In: *ACM transactions on graphics (TOG)*. Vol. 25. 3, pp. 835–846.

Snyder, Jeff (2010). “Doppler Velocity Log (DVL) navigation for observation-class ROVs”. In: *MTS/IEEE OCEANS, SEATTLE*, pp. 1–9.

Stone Aerospace (Apr. 2015). *Digital Wall Mapper*. URL:`http://stoneaerospace.com/digital-wall-mapper/`.

Stone, W. C. (2007). “Design and Deployment of a 3-D Autonomous Subterranean Submarine Exploration Vehicle”. In: *International Symposium on Unmanned Untethered Submersible Technologies (UUST)*. 512.

Strasdat, Hauke (2012). “Local accuracy and global consistency for efficient visual SLAM”. PhD thesis. Citeseer.

Sturm, J. et al. (2012). “A Benchmark for the Evaluation of RGB-D SLAM Systems”. In: *Proc. IROS*, pp. 573–580.

Sun, K. et al. (2018). “Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight”. In: *IEEE Robot. Autom. Lett.* 3.2, pp. 965–972. ISSN: 2377-3766. DOI: `10.1109/LRA.2018.2793349`.

Tarrio, Juan José and Sol Pedre (2017). “Realtime Edge Based Visual Inertial Odometry for MAV Teleoperation in Indoor Environments”. In: *J Intell. Robot. Syst.* Pp. 235–252. DOI: `10.1007/s10846-017-0670-y`.

Umeyama, Shinji (1991). “Least-Squares Estimation of Transformation Parameters Between Two Point Patterns”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 13.4, pp. 376–380. ISSN: 0162-8828. DOI: `10.1109/34.88573`.

Weidner, Nicholas et al. (May 2017). “Underwater Cave Mapping using Stereo Vision”. In: *Proc. ICRA*. Singapore, pp. 5709–5715.

Wirth, Stephan, Pep Lluis Negre Carrasco, and Gabriel Oliver Codina (2013). "Visual odometry for autonomous underwater vehicles". In: *OCEANS-Bergen, 2013 MTS/IEEE*. IEEE, pp. 1–6.

Wu, Changchang (2013). "Towards linear-time incremental structure from motion". In: *2013 International Conference on 3D Vision-3DV 2013*. IEEE, pp. 127–134.

Wu, X. et al. (2015). "Towards a Framework for Human Factors in Underwater Robotics". In: *Human Factors and Ergonomics Society International Annual Meeting*, pp. 1115–1119.

Xu, Zexuan et al. (Aug. 2016). "Long distance seawater intrusion through a karst conduit network in the Woodville Karst Plain, Florida". In: *Scientific Reports* 6, pp. 1–10.