

Development, Evaluation and Validation of a Stereo Camera Underwater SLAM Algorithm



Marc Hildebrandt
hildebrandt.marc@gmx.de
University of Bremen

A thesis submitted for the degree of
PhilosophiæDoctor (PhD) of Enigneering
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

2013 December

1. Reviewer: Prof. Dr. Frank Kirchner

2. Reviewer: Prof. Dr. Joachim Hertzberg

Day of the defense: 12.03.2014

Abstract

In this work the development of an algorithm for visual underwater localization is described. It spans the complete process from the initial idea, the development of a suitable underwater vehicle for testing to the algorithm's experimental validation in real underwater environments. Besides the development and validation of the visual SLAM algorithm, the methodology for its evaluation is a key aspect of this work. The resulting SURE-SLAM algorithm uses a stereo camera system and basic vehicle sensors (AHRS, DPS) to compute a complete, error-bounded localization solution for underwater vehicles in real-time with similar quality as state-of-the-art acoustically stabilized dead-reckoning approaches. The robustness of the algorithm as well as its limitations and failure-cases are established by extensive field testing with the AUV DAGON, which was developed during this thesis as test and evaluation vehicle.

Acknowledgements

I would like to thank my family, especially my wife Vera and my son Jonathan for their patience and support while working on this thesis, as well as our dog Fiocca for the serenity she emanates.

Further I would like to thank all my colleagues who helped me shaping and improving this thesis and without whom this work could not have been concluded in this fashion.

Special thanks go to Christopher Gaudig, who supported me in virtually all experiments with DAGON, as well as Marius Wirtz for his invaluable help in integrating and maintaining the vehicle.

Many of the photographs shown in this work have been taken by Dr. Jan Albiez, who accompanied many of the experiments as senior scientist and advisor.

Parts of this work were conducted during the CUSLAM-project, funded by the German Federal Ministry of Economics and Technology (BMWi) according to a resolution of the German Bundestag, grant no. 03SX290.

Contents

List of Figures	xi
List of Tables	xvii
Glossary	xix
1 Introduction	21
1.1 Motivation	21
1.2 State of the Art	23
1.2.1 Underwater Vehicles	23
1.2.2 Underwater Localization/Navigation	24
1.2.2.1 Dead-Reckoning	24
1.2.2.2 Acoustic Localization	25
1.2.2.3 Visual Localization	25
1.2.3 Problems of Existing Approaches	28
1.3 Problem Description	28
2 Methodology	31
2.1 Development Process	31
2.2 Validation	33
2.2.1 Terminology of Underwater Localization	33
2.2.1.1 Reference Localization	34
2.2.2 The Necessity of a Validation Process	34
2.2.3 The Gold-Standard Method for Localization	37
2.2.4 Validation Process and Performance Metrics	38
2.2.4.1 Selection and Implementation of GSRL	39

CONTENTS

2.2.4.2	Sparse Validation of GSRL Implementation	39
2.2.4.3	Implementation of NLA	39
2.2.4.4	Recording of Validation Data	39
2.2.4.5	Performance Metrics	39
2.3	Real-Time Constraints	41
3	Design of a Versatile AUV for High Precision Visual Mapping and Algorithm Evaluation	43
3.1	Introduction	43
3.1.1	Decision to Build a New AUV	44
3.1.2	External Help	47
3.2	Design Criteria	47
3.2.1	Mechanical Requirements	47
3.2.2	Sensor/Instrument Requirements	48
3.2.3	Vehicle Concept	49
3.3	System Description	52
3.3.1	Pressure Hull	53
3.3.2	Battery	54
3.3.3	Data Processing and Communications	55
3.3.4	Fiber-Optic Cable	57
3.3.5	Electronic Fuse	58
3.3.6	Thrusters	58
3.3.7	Vision System	59
3.3.8	Reference Localization System	60
3.3.9	Basic Controllers	61
3.3.10	Station-Keeping	61
3.3.11	Trajectory-Follower	62
3.4	Operation	62
3.4.1	Testing Environments	63
3.4.2	Field Testing	63
3.5	Evaluation	64
3.5.1	Future Work	64

4	Algorithm	65
4.1	Reference Localization Implementation	65
4.2	SURE-SLAM Algorithm	66
4.2.1	Visual Odometry	68
4.2.1.1	Pre-Processing	68
4.2.1.2	Feature Extraction	69
4.2.1.3	Projection into 3D-Space	70
4.2.1.4	Image Homography/FM Estimation	70
4.2.1.5	ICP Optimization	72
4.2.2	SLAM-Backend	73
4.2.3	Loop-Close Detection	74
4.2.4	Performance and Optimizations	76
4.2.4.1	Runtime Analysis	76
4.2.4.2	GPU Optimizations	77
4.2.4.3	Multi-Core Optimizations	79
4.2.4.4	Memory requirement	81
5	Experiments	83
5.1	Testing Environments	83
5.1.1	Synthetic Data	83
5.1.2	Space Exploration Hall	85
5.1.3	Black Basin	86
5.1.4	Glass Tank	86
5.1.5	University Pool	87
5.1.6	Unisee	87
5.1.7	Kiel	88
5.1.8	Rostock	89
5.1.8.1	LBL-failure	90
5.2	Description of Individual Experiments	90
5.2.1	Meander without Crossings (Synthetic)	91
5.2.2	Spider-Cam (Space Exploration Hall)	93
5.2.3	Camera Calibration (Glass Tank)	94
5.2.4	Turbidity (Black Basin)	95

CONTENTS

5.2.5	Repetitive Environment (University Pool, Norway)	96
5.2.6	Sparse Validation of Reference Measurement (Glass Tank)	97
5.2.7	Sparse Validation of Reference Measurement (Unisee)	99
5.2.8	Indoors Validation (Glass Tank)	100
5.2.9	Dynamic Environment (Kiel)	102
5.2.10	Under Ice (Unisee)	104
5.2.11	Outdoors Validation (Rostock)	105
5.2.12	Brute-Force Loop-Closing (Rostock)	106
5.2.13	Long-Term Stability (Glass Tank)	107
5.2.14	Sparse Environment (Synthetic)	108
5.2.15	Impact of Internal Navigation Hardware on NLA quality (Glass Basin)	111
5.2.16	SLAM vs. Visual Odometry (Unisee)	113
5.2.17	Sherpa (Space Exploration Hall)	114
5.2.18	Mosaicking (Glass Basin)	115
6	Conclusion	119
6.1	Description of Failure Cases	119
6.2	Resulting Characteristics for GSRL	121
6.3	Resulting Characteristics for NLA	121
6.4	Comparison Between GSRL and NLA	122
6.5	Future Work	122
6.5.1	Future of the SURE-SLAM Algorithm	123
6.5.2	Extensions of the SURE-SLAM Algorithm	125
6.5.3	Future of the AUV Dagon	127
	References	129

List of Figures

1.1	The SeaBotix LBV 150-2 in an outdoor test in Denmark, Summer 2007 (a). Dry view of the LBV 150-2 (b).	22
1.2	Two AUV classes extensively used in localization research: A seaBED class AUV by WHOI (a), a DEPTHX/ENDURANCE class AUV by NASA (b).	24
2.1	Incremental development process.	32
2.2	Vehicle’s body-fixed frame coordinate system.	33
3.1	The AUV “AVALON”.	45
3.2	Parallel setup without peripheral equipment.	50
3.3	The AUV DAGON evolving: pressure hull testing (a); basic indoor testing (b); outdoor testing (c); current state (2013). (d).	52
3.4	Head mechanics with camera and fixture for power electronics (a); Rear caps and quick fasteners (b).	54
3.5	Data and power connections within the vehicle.	56
4.1	Flow-chart of the GSRL implementation on the AUV DAGON.	66
4.2	Flow-chart of the NLA implementation on the AUV DAGON.	67
4.3	The main steps in the visual odometry algorithm.	68
4.4	The basic idea of a graph-based SLAM approach.	69

LIST OF FIGURES

4.5	The relations inside two subsequent image pairs. The extracted features are shown as blue circles. The stereo correspondences are shown as red lines (only for the lower pair), which are horizontal since the images are rectified. The green lines show the inter-frame correspondences (only shown for the left image pair), which are vertical since the cameras moved in a straight line between the two frames.	71
4.6	Loop Closing. Brute-force approach with n matching candidates (a). Graph-based approach with only 3 matching candidates (b).	74
4.7	Performance of SURF extraction on different generation of CPUs and GPUs applied to differently sized images.	78
4.8	Processing pipeline for visual odometry.	79
4.9	The location of all detected and stereo-matched features on all left images of the Rostock test sequence (approx 4.500 images) plotted as greyscale image. Dashed red line shows the cropping limit (130 pixels).	80
5.1	A sequence of images from the test dataset showing one of the u-turns. It shows 20 frames from the left camera, using every 10th frame.	83
5.2	The scene used to create the synthetic dataset. Note that an additional omni directional light was added to create this view of the scene.	84
5.3	Computer rendering of the space exploration hall at the DFKI RIC (a). View of the crater with Spider-Cam and Kuka-satellite-simulator (b).	85
5.4	A view of the DFKI-RIC's underwater testbed with the glass basin in the foreground and the black basin at the back.	86
5.5	DAGON in the University Pool 2010 (a). Camera view of the floor of the university pool. The camera's distortion is well visible in the tile pattern (b).	87
5.6	The lake near the university, called "Unisee". Satellite view of the lake with depth profile (a). DAGON surfacing at the end of an experiment in the evening (b).	88
5.7	Deployment of DAGON in the harbor in Kiel, Winter 2011/2012 (a). Nighttime experiment with DAGON visible as bright area at the right border of the photo (b).	88

5.8 Team photo of the sea-trials summer 2012 in Rostock, taken on the R/V <i>Gadus</i> on the last day of the trials (a). The testing environment at the artificial reef Nienhagen (54°10.5'N 11°56,6'E). To the left the observation tower can be seen, to the right the research vessel <i>Gadus</i> , which was used as base (b).	89
5.9 Experiments at the artificial reef in Rostock, Germany Summer 2012: Deployment of the LBL system from a small RBI (a). The vehicle in the water (b).	90
5.10 Experiments at the artificial reef in Rostock, Germany Summer 2012: deployment of the vehicle from the support ship (a). Preparation of the vehicle and LBL system on deck in the left figure (b).	91
5.11 Distance error between visual odometry and ground truth for local and global algorithm variants (a). Comparison between the performance of visual odometry with and without a global map (b).	92
5.12 Geometric relations of two consecutive stereo pairs with features recorded during the spider-cam trials (a). Reconstructed trajectory of the spider-cam trial with evaluation of low-quality regions (b).	93
5.13 Results of camera calibration in the underwater testbed.	94
5.14 Measurements with increasing image turbidity (a). Effect of turbidity in a real environment. The floor in this image is slanted, the approximate image distance in the lower-right corner is 1.5 m while in the upper-left corner it is 3 m (b).	95
5.15 Camera view of the fish net in Norway.	97
5.16 Position estimation by the GSRL for the sparse indoors validation dataset. Ten consecutive squares (2.5x2.5 m) with manually enforced start- and end positions. Due to the confined space of the test-tank, an obstacle-avoidance behavior was active at the same time, accounting for the offset of squares.	98
5.17 X, Y, and Z components of the position estimation of the GSRL during the sparse indoors validation dataset.	99
5.18 Position estimation by the GSRL for the sparse outdoors validation dataset. One large square (100x100 m) with manually enforced start- and end positions.	100

LIST OF FIGURES

5.19	Trajectories as estimated by the SURE-SLAM for the indoors validation dataset.	101
5.20	X-component of the first loop of the trajectory with standard deviation of GSRL and NLA of the indoors validation dataset.	102
5.21	Estimated trajectory with LBL readings in the Kiel harbor (fall 2011). .	103
5.22	Under-ice trials winter 2011/12. Sample frame of under-ice dataset with unsuccessful inter-frame correspondences due to non-Lambertian surface (a). The vehicle surfacing after a successful trajectory (b).	104
5.23	Trajectories as estimated by the GSRL and SURE-SLAM for the outdoors validation dataset.	105
5.24	The trajectory of the brute-force loop-closing experiment with loop-closure locations.	107
5.25	Visual data from the Rostock datasets. Inter-frame-matching of image-features (a). Loop-closing match of image features with a time-delay of 30 minutes (b).	108
5.26	Trajectory as estimated by the GSRL for the long-term hovering experiment. After starting in one corner of the glass tank the vehicle was hovering for 90 min in the center, and then returned to the initial position.	109
5.27	A combination of the nine synthetic scenes used for testing in a variably sparse environment. The vehicle trajectory is shown in red.	110
5.28	Reconstructed trajectories from the sparse synthetic datasets. Complete set of loop closures in the low ruggedness/high texture dataset (a). No detected loop closures but working visual odometry in the medium ruggedness/medium texture dataset (b).	111
5.29	Impact of INH (DPS, AHRS and FOG) on NLA quality with corrupted camera calibration and increased vehicle speed. INH de-activated, visual odometry only.	113
5.30	Impact of INH (DPS, AHRS and FOG) on NLA quality with corrupted camera calibration and increased vehicle speed. IHN activated, orientation drift compensated.	114
5.31	Impact of loop closing on trajectory: only visual odometry has a large gap in the trajectory, the SLAM trajectory has closed that gap by loop closing.	115

LIST OF FIGURES

5.32	The Sherpa robot with its stereo camera system surveying the ground (a). Geometric relations of two consecutive stereo pairs with features recorded by the Sherpa robot (b).	116
5.33	Mosaic created from images and correspondences of the glass basin environment.	117
6.1	Topside view of the new exploration basin (a). View through the lower viewport into the basin (b).	123
6.2	Schematic overview of a possible mission scenario for the Europa-Explorer project. 0) ice-drill penetrated the ice-shield. 1) AUV has been released from the payload compartment. 2) AUV descends to ocean floor. 3) exploration using cameras/sonar and internal sensors. 4) ascend to the ice/water boundary. 5) return to ice-drill (using autonomous localization buoys) and docking for energy/data exchange.	124

LIST OF FIGURES

List of Tables

3.1	List of sensors and instruments of the AUV.	53
4.1	Average computation times for the major steps of the un-optimized visual odometry algorithm, 640x480 input image.	77
4.2	Processing speed for feature extraction on a 960x540 stereo image pair on different hardware with different optimizations.	81
5.1	Effect of turbidity to feature stability.	96
5.2	Results of the sparse environment experiments. The first value in each cell is the feasibility of the terrain for visual odometry, the second value the number of detected loop closures.	112
6.1	Performance results	120
6.2	Comparison between the SURE-SLAM and the GSRL. All values are upper limits/upper bounds.	122

LIST OF TABLES

Glossary

AHRS	Attitude-Heading Reference System
AUV	Autonomous Underwater Vehicle
AVALON	Autonomous Vehicle for Aquatic Learning, Operation and Navigation
BA	Bundle Adjustment
CCD	Charge-Coupled Device
COG	Center of Gravity
COTS	Commercial off the Shelf
CUSLAM	Confined Underwater SLAM
CWDM	Coarse Wavelength Division Multiplex
DFKI	Deutsches Forschungszentrum für Künstliche Intelligenz - German Research Institute for Artificial Intelligence
DOF	Degree of Freedom
DPS	Digital Pressure Sensor
DVL	Doppler Velocity Log
ECEF	Earth-Centered Earth-Fixed
EKF	Extended Kalman Filter
FOG	Fiber-Optic Gyroscope
FOV	Field of View
FPGA	Field Programmable Gate Array
FPS	Frames per Second
FTU	Formazin Turbidity Units

GLOSSARY

G²O	General Graph Optimization
GPU	Graphics Processing Unit
GSRL	Gold-Standard Reference Localization
HDD	Hard Disk Drive
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
INH	Internal Navigation Hardware
KF	Kalman Filter
LBL	Long Base-Line
NLA	New Localization Algorithm
OpenCV	Open Computer Vision
PWM	Pulse-Width Modulation
RANSAC	Random Sample Consensus
RIC	Robotics Innovation Center
ROCK	Robotics Tool Kit
ROI	Region of Interest
ROV	Remotely Operated Vehicle
RPM	Rounds per Minute
RV	Research Vehicle
SBL	Short Base-Line
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURE-SLAM	Stereo Underwater Realtime Exploration-SLAM
SURF	Speeded-Up Robust Features
TDP	Thermal Design Power
USBL	Ultra-Short Base-Line
VAN	Visually Aided Navigation
VHDL	Very High Speed Integrated Circuit Hardware Description Language

1

Introduction

1.1 Motivation

The original motivation for this work was the necessity to acquire the ability to conduct missions of varying complexity with underwater vehicles. With a number of early experiments and trials with the ROV “LBV 150-2” (see figure 1.1) and the AUV “AVALON” (shown in figure 3.1 in chapter 3) it became apparent, that the ability to localize the vehicle is a key requirement for execution of virtually all tasks. This fact was at the time greatly underestimated by the author and the underwater robotics group at the DFKI-RIC. The intended solution for this problem was the acquisition of an acoustic USBL tracking system by Tritech (MicroNAV). Both vehicles could be equipped with the conveniently small transponders, the operational overhead was relatively small. Unfortunately, the system could provide neither the information necessary for most tasks, nor the quality required. In the environments in which it was tested (basin, lake, harbor) it was prone to drop-outs, signal jumps and coarse measurements. Additionally it was unable to provide a heading measurement, even with the usage of three transponders on the vehicle (11). These experiences made it very clear that a vehicle-centered, globally consistent localization approach was required. After a survey of the available solutions both on the market and scientifically, two candidates for such a system could be identified: an LBL-stabilized, DVL-based dead-reckoning system on the one hand, and a vision-based SLAM system on the other hand. Both systems would require implementation of localization and navigation software, with different focus on sensor processing and image processing. Since the visual approach promised even more

1. INTRODUCTION



Figure 1.1: The SeaBotix LBV 150-2 in an outdoor test in Denmark, Summer 2007 (a). Dry view of the LBV 150-2 (b).

independence from infrastructure (no LBL required) and was deemed cheaper (more implementation effort but less required sensors), the decision for a visual localization algorithm was reached.

At this point I was working for two years at the DFKI-RIC’s underwater robotics department and involved in most of the work described above. I had mainly worked on the CManipulator project ¹, in which visual manipulation with an underwater hydraulic manipulator utilizing a stereo camera system was researched. Due to my interest in mobile underwater robots I had created the “AVALON” AUV within a undergraduate student’s project in parallel to this work. When it became apparent that a visual solution would be required I started writing a proposal for a thesis based on the requirements described above. Unfortunately, two main problems arose: The vehicle AVALON would not be sufficient for the experiments required for such a thesis. The second problem was validation. All methods for validation of localization algorithms for underwater vehicles described in literature (for further details see section 1.2) did not seem adequate in characterizing an algorithm. Ideally a ground-truth should be available as gold-standard against which the visual algorithm could be compared and tested. The question how to acquire such measurements in real underwater environments lead back to the LBL-stabilized, DVL-based dead-reckoning system. This again

¹<http://robotik.dfk-bremen.de/de/forschung/projekte/cmanipulator.html>

would mean that I would need to build a vehicle equipped with both sensor systems, the visual system for my algorithm and the acoustic sensors for the reference localization. I drafted a proposal including estimated costs for such a system and went to a funding agency. After some modifications, they approved of the concept and asked the DFKI-RIC to create a fully-fledged project out of it. This led to the CUSLAM-project, in which most of the work described in this thesis was conducted.

1.2 State of the Art

The following section will describe the current state of the art with respect to this work in two categories: a brief review of underwater vehicles, and a more in-depth review of localization and navigation algorithms for underwater vehicles.

1.2.1 Underwater Vehicles

The SeaBED AUV class (see figure 1.2(a)) was developed by the Woods-Hole institution in the early years of the millennium, and has since then been put to extended use. Specifically designed as vehicle for close-range tasks such as side-scan bathymetry and visual bathymetry (51), it was a new approach in the mid-size AUV class. Its extensive sensor suite as well as excellent roll-stability make it the ideal vehicle for tasks which require detailed data of small underwater areas, such as underwater archeology (56). This AUV class is used extensively by a number of laboratories world-wide, including the National Sun Yat-Sen University in Taiwan and the Australian Center for Field Robotics at the University of Sydney.

NASA developed the DEPTHX AUV (DEep Phreatic THERmal eXplorer, see figure 1.2(b)) as an evaluation platform for navigation during future missions on Jupiter's ice-moon Europa (22). On earth it was tested in a number of hydro-thermal vents and caves. With its 2 m diameter and weight of 1.35 t it easily falls into the large AUV class. Besides a standard sensor suite for acoustic dead-reckoning (see section 1.2.2.2) it incorporates 32 single-beam sonar transducers distributed along its hull for its unique sonar-based SLAM localization approach. Specifically designed for confined quarters, it performs very well as long as a large number of the sonar-beams provide an echo (19). Since 2006 the vehicle was continued under the name ENDURANCE (Environmentally Non-Disturbing Under-ice Robotic ANTarctic Explorer) by Stone

1. INTRODUCTION

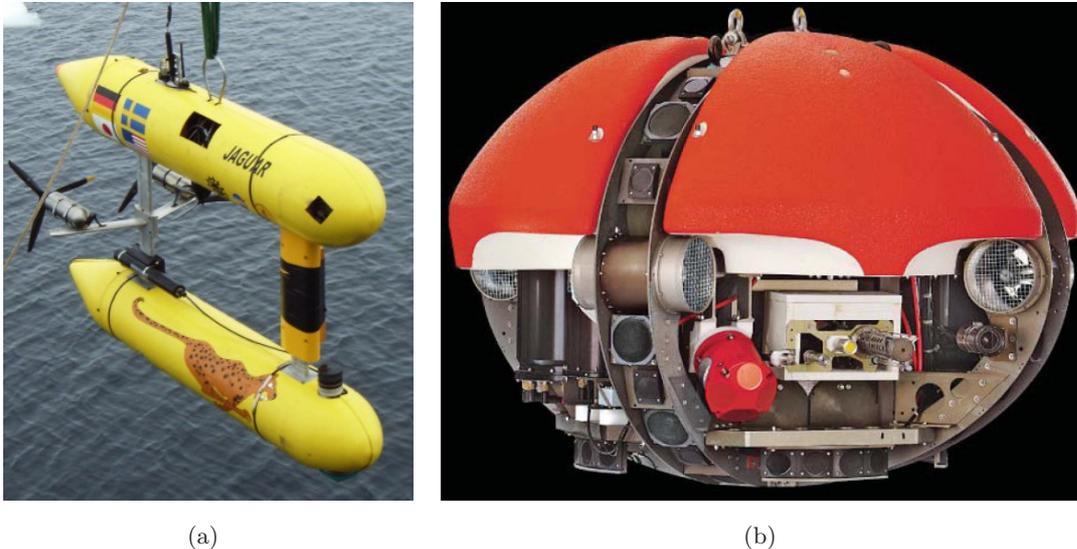


Figure 1.2: Two AUV classes extensively used in localization research: A seaBED class AUV by WHOI (a), a DEPTHX/ENDURANCE class AUV by NASA (b).

Aerospace, specifically addressing the vehicle’s open-area navigation capability with the addition of visual navigation and an USBL beacon system (44).

1.2.2 Underwater Localization/Navigation

Underwater localization can be categorized by the type of sensor it used. Three main types are common: dead-reckoning, acoustic localization and visual localization.

1.2.2.1 Dead-Reckoning

Dead reckoning is a method for localization that only relies on vehicle-internal sensors. Classical dead-reckoning uses time measurement, the vehicle’s orientation and an estimate of speed to calculate a vehicle’s current position. While time and orientation can be measured easily enough (compass or AHRS), vehicle speed is more complicated, and usually estimated using a Pitot-tube, which measures the vehicle’s speed relative to its surrounding water (32). An alternative to the direct measurement of vehicle speed is the usage of a navigation grade IMU. Using an IMU the vehicle’s accelerations are accumulated, resulting in an indirect measurement of speed. The drawback of this method is that only very precise and thus expensive and cumbersome devices

can provide suitable measurement qualities for this setup. The big advantage of a self-contained dead-reckoning is its independence of the surroundings. Its main drawback is that its long-term error is un-bound, and in practical terms strongly dependent on both AHRS and IMU accuracy (34).

1.2.2.2 Acoustic Localization

Acoustic localization can be considered the current standard in underwater robotics. In addition to the basic dead-reckoning sensors as described above, a DVL sensor is included. A DVL measures speed over ground by measurement of the doppler-shift of an acoustic signal which bounces off the sea floor. Properly used this can provide a very good estimate of speed-over-ground for the dead-reckoning filter, resulting in an acoustic-dead-reckoning. Drawback of this approach is the necessity to remain in proximity to the sea floor, since a DVL does not have unlimited range (usually 10 m to 500 m, depending on the device). While usually more stable than pure dead-reckoning, the restriction of un-bound long-term error still is relevant. In order to remedy this, beacon based localization is included in the localization setup. A beacon based localization device uses triangulation to measure the position of a mobile transponder relative to three or more stationary transponders (respectively one stationary transponder for USBL). Depending on the distance of the stationary transponders, these systems are called LBL, SBL or USBL systems (Long Base-Line, Short Base-Line or Ultra-Short Base-Line). The individual techniques differ mostly in complexity of set-up and accuracy (48), (6). While requiring prior setup and line-of-sight during missions, these systems provide drift-free localization information. This information is usually less accurate than necessary for precision missions, but it is accurate enough to provide an upper bound for the long-term error when used in conjunction with an (acoustic) dead-reckoning system.

1.2.2.3 Visual Localization

In visual underwater localization a camera system provides localization data. This data can be either used as sole source for localization, or in combination with dead-reckoning or acoustic data.

One of the best documented work in the field of underwater visual navigation is the work of Eustice. In his PhD thesis (16) he introduced the concept of VAN, visually

1. INTRODUCTION

aided navigation. This approach used the camera available on the AUV to improve its navigation capabilities. Its superiority over acoustic dead-reckoning-only methods was shown when they surveyed the RMS Titanic (17). Due to the large distance to the surface, the USBL sensor used to stabilize the dead-reckoning navigation systems (DVL, IMU, pressure) experienced strong deviations, which could be eliminated by the incorporation of data from the visual system. As the name of the approach implies, it is neither meant nor suited as replacement of an acoustic dead-reckoning navigation system, but only as an enhancement for post-processing.

Brown (9) extended the VAN approach to incorporate a second camera, forming a stereo camera system. Using the new visual information of distance to the ground, a vision-only SLAM was possible. In their application they still used the other available vehicle sensors (DVL, AHRS, pressure) to perform high-accuracy hull inspections. This work was extended on by Kim (33) in 2013 with the addition of a saliency metric to improve loop closing fidelity. No reference localization data was recorded.

Mahon et al. (39) describe a VAN-based approach using a stereo-vision system for loop closure hypothesis formation and matching in a post-processing approach. Unfortunately, they do not have any reference localization, only a GPS-location of start/end after diving/surfacing in a 40 m deep environment.

In his Ph.D. thesis Richmond (49) describes an online approach for navigation and mosaicking, but most information for the localization originates from vehicle sensors like DVL and FOG: “Vision is only used where it is most effective, complementary sensors whose data require much less processing (such as DVL, altimeter, and orientation sensors) are used whenever possible”. The vision processing uses the orientation and distance data from the vehicle sensors to compute the infinite homography for each image, putting them into a normalized form prior to any vision processing. This makes the use of a relatively simple SLoG-Filter (46) possible for inter-frame matching and correlation, explaining the real-time capabilities of this approach. No reference measurement for his results is presented.

A very impressive example of how the data from such visual surveys can be utilized is given by Johnson-Roberson (31). Here the VAN approach is used as basis for fully 3D reconstruction of a sea-floor survey. The accuracy and consistency of the resulting bathymetrical data shows the potential of these visual-aided approaches - with

the drawback that the method is designed as post-processing step with no real-time capabilities.

In 2008 Salvi and Thomas presented a stereovision-only SLAM approach for underwater vehicles (53), (60). Using the images from a stereo camera system on an AUV, stable feature estimators (SURF and SIFT) are extracted from the stereo images to create highly salient 3d landmarks. These landmarks are matched between stereo pairs to calculate the images' fundamental matrix and thus camera motion. When used on video data, a visual odometry is the result. After they could only verify their results using simulated data, in subsequent publications its feasibility for real-world environments could be verified (4) without going into details on the real-time capabilities.

Salvi's approach was improved by Aulinas (2) with the addition of a pre-processing step. In this step, the camera images are searched for regions of interest (ROIs) before the feature extraction. The rationale is that if such a region of interest exists, it will provide more salient features for further processing than just using the whole image as feature source. the ROIs are computed by edge-detection and hue-channel filtering.

Corke et al. (13) describe a series of experiments where visual odometry based navigation on a small Starbug AUV is compared with the results from a LBL tracking system. While the visual odometry algorithm is fairly basic (described in more detail in (15)), the idea of using a system yielding dual measurements which they then compare in a later stage is very promising. Their 'ground truth' measurement (reference measurement in this work's nomenclature) is done by a drift-free system (LBL), so the accuracy of their visual odometry can be quantified quite well. For their experiments it lay in the order of 5 m after 100 m travel or 5 % of the traveled distance. Their approach does not incorporate any measure to create an upper bound for this error.

Milella and Siegwart (41) describe a basic framework for stereo camera motion estimation using iterative closest point (ICP) approaches. Using a feature descriptor to select salient points in a stereo image pair, tracking them across consecutive image pairs and using ICP for inter-frame matching they achieve reasonable results for land based robots. Even though their application scenario was land based robotics, the idea of combining stereo cameras with ICP approaches strongly influenced this thesis.

Moreno et al. (43) describe a visual odometry framework based on a stereo camera system. Their work does not incorporate any other sensors, yet the basic idea is similar to the approach of the visual part presented in this thesis: selecting features

1. INTRODUCTION

(Lowe’s scale invariant feature transform (SIFT) (38) in this case) in a stereo pair, re-projecting them into 3D space using epipolar constraints, tracking the features over a number of frames and using the gathered information in a probabilistic approach for pose change estimation. They propose their work as input for a SLAM approach, since no global reference map or loop closing approaches are employed and thus their algorithm’s performance degrades strongly with respect to driven distance.

1.2.3 Problems of Existing Approaches

One of the main problem of acoustic approaches was already mentioned: The necessity for an infrastructure system (LBL, USBL) to provide long-term stability of localization. In the best-case scenarios this results in major overhead during practical application of such localization systems, in worst cases this makes such approaches non-feasible (e.g. acoustic-disabled zones, in-structure surveys). A big advantage of these approaches is their inherent redundancy: When only the dead-reckoning or the global localization fails for short periods, this is automatically compensated (with the penalty of reduced accuracy in the meantime).

There are two major problems with existing visual approaches. The first issue is the lack of a stringent validation of results. This is arguably a tough task in the demanding underwater environment where a “ground-truth” measurement is impossible to achieve. Despite this, the practice of omitting it altogether is considered problematic. The second issue is the fact that vision systems are considered sensors of secondary quality. Most approaches only use vision as complementary or supporting sensor data, denying its fitness as equal to the established inertial/acoustic sensors. While having a number of problems, limitations and issues it is the author’s believe that with careful implementation vision can be considered to be on the same level as other sensors when it comes to robustness, measurement quality and usability.

1.3 Problem Description

The aim of underwater localization is to find the vehicle pose $\eta = [x, y, z, \phi, \theta, \psi]$ in 6 DOFs, consisting of the vehicle position $\eta_1 = [x, y, z]$ and its Euler orientation $\eta_2 = [\phi, \theta, \psi]$. The Earth-centered Earth-fixed (ECEF) frame $\{e\}$ is used as the reference frame. Measurement of these quantities is of different complexity: z, ϕ, θ are

relatively easy to determine in the underwater domain: z can be absolutely measured with pressure sensors as the distance between the the surface and the vehicle. ϕ (roll) and θ (pitch) can be measured by simple AHRS sensors. AHRS sensors usually employ accelerometers, gyroscopes and magnetometers to compute a 3d orientation. As long as the vehicle is not strongly accelerating, earth's gravity can be used as stabilizing quantity for roll and pitch to make these two measurements drift-free and readily usable. The other quantities, x, y and ψ do not have these benefits. It can be argued that measurement of ψ can be achieved by AHRS sensors as well, using earth's magnetic field as stabilizing quantity to prevent drift. Since the magnetic field is much easier distorted than gravity (by large metallic objects, vehicle electronics and thrusters, natural misguiding) and may not be available (e.g. polar regions) it is not regarded as safe stabilization quantity. Stable measurement of x, y and ψ thus becomes the main problem to be solved.

The following properties were tried to be reached with the algorithm developed in this work:

Local Measurements The vehicle pose η is to be measured by a non-infrastructure localization system. This excludes all acoustic reference measurement systems, since they require the existence of pre-setup localization beacon(s), which is to be avoided to increase versatility. Other infrastructure, be it special visual/acoustic markers, surface buoys or ground structures are not to be used. Ideally, mainly visual sensors should be used. This aim does not exclude the usage of other sensors, especially an AHRS and a pressure sensor as described above, since they are usually part of most underwater vehicles, even simple ones.

Error Estimation The measurement system should estimate its own localization error. This error should be independent of mission time and traveled distance. It should be possible to determine an upper bound for the localization error which is not violated during typical missions.

Independence of Environment The measurement system should be as versatile as possible. This especially means that only minimal conditions for the environment are necessary. Since the measurement system is image based, this particularly means that

1. INTRODUCTION

the requirements for ground structure are very low. The terrain and ground texture should not need to be known beforehand, since this would severely limit the algorithm's usability. While it will not be possible to formulate an algorithm which can cope with every environment, the known limitations should be tested and well formulated.

Real-Time Capabilities The last requirement is the algorithm's real-time capability. Since it is to be used as localization system on a live vehicle, it needs to be able to compute η while the vehicle is conducting its mission.

2

Methodology

2.1 Development Process

One of the aspects which stood at the beginning of this thesis was the idea to do a stepwise development and validation of the algorithm. This is deemed necessary, since an attempt of development and implementation of a complex algorithm on noisy real-world data is not recommended. An additional reason was that during the first development steps the AUV DAGON (for more information on the vehicle see chapter 3) was not yet available for testing. So the first step in development was to create a plan for the validation steps:

- Synthetic simulation data

For the first steps of implementation it is crucial to have the best quality of ground-truth data available. Additionally a great number of specific datasets for testing are required. This makes a simulated environment yielding synthetic data the ideal candidate. Realistic rendering tools (such as 3DStudioMAX) can create adequate scenes with high-quality lighting and camera behavior. Ground-truth is easily available through knowledge of the selected trajectories and image properties, the complexity of the data freely selectable. The downside of the data is obviously the abstractions made by the rendering tool and the inability to create all aspects of real world data.

- Lab experimental data

For the next steps it is important to migrate from simulated data to real camera

2. METHODOLOGY

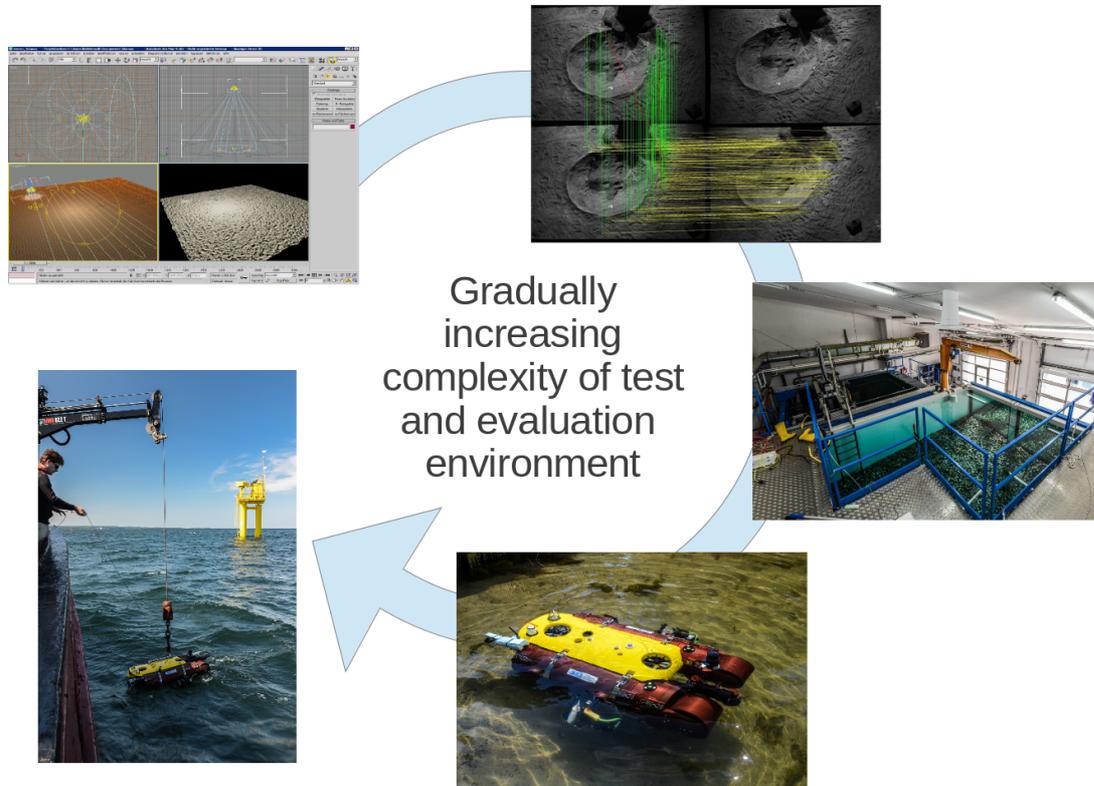


Figure 2.1: Incremental development process.

data. This adds a lot of small differences which are easily omitted in simulation. This includes camera distortion, mis-calibration, shadows, exposure and camera triggering to name a few. Another big step is the usage of underwater camera data, since most of the aspects named have even more impact in this environment. The advantage of lab data is that it is acquired in semi-controllable environments, with the ability to reduce (or specifically select) a number of environmental conditions: external lighting, turbidity, structure of environment, distance to objects. Ground-truth of camera motion is more complicated but still computable, by the use of guided systems (gantry crane, external tracking system, SpiderCam).

- Real-world data

At some point the algorithm needs to be tested in its designated environment. No amount of simulation or lab testing can ever replace this step, since only there unforeseen conditions, un-modeled parameters or unexpected sources of noise

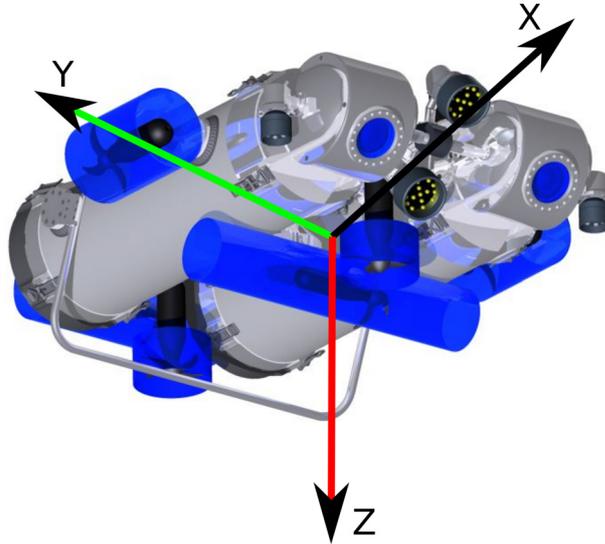


Figure 2.2: Vehicle’s body-fixed frame coordinate system.

can be identified. Acquisition of this type of data requires major preparation and resources, as it requires the use of an outdoor underwater robot, and thus is sensitive to weather, sea-state, selection of test-site and hardware failure. For this type of data to be usable, a method to record reference localization data is required, since the vehicle usually is not directly visible to the operator and thus its trajectory can be very hard to estimate.

2.2 Validation

2.2.1 Terminology of Underwater Localization

A frame as shown in figure 2.2 is attached to the vehicle’s origin and used as body-fixed frame. The vehicle’s position and its orientation are expressed in ECEF frame and its linear and angular velocities are expressed in body-fixed frame. Each measurement of the pose is accompanied with a covariance, giving a measure of uncertainty. Ideally the transformation of the world coordinate system to the geographic coordinate system is known. Since this is not an intrinsic necessity for comparison of underwater localization techniques and since not all approaches will yield this transformation, for the rest of this thesis the geographic coordinate system is not used.

2. METHODOLOGY

Methods for underwater localization can be categorized in multiple ways - this paper will use the actor-based categorization. It differentiates between infrastructure-centric localization and vehicle-centric localization. Infrastructure-centric localization uses an external localization system, which has to be set-up in the mission area prior to any further work. Examples for infrastructure localization techniques are beacon systems like USBL, SBL and LBL as described in section 1.2.2.2. Vehicle-centric localization uses only the sensors built into the vehicle to be localized. These sensors can either interact with the environment (exteroceptive) or only measure internal parameters (interoceptive). An example for a vehicle localization technique combining both types is dead-reckoning using an IMU (inertial measurement unit) and a DVL (Doppler velocity log).

2.2.1.1 Reference Localization

In this work the term reference localization is used as term for a second, “Gold-Standard” localization technique. The term “ground-truth”, which is often used to describe such a measurement is believed to be misleading, since it hides the uncertainty every measurement in the real world is afflicted with. For this reason the term reference localization is used for all real-world measurements, while ground-truth is reserved for simulated or synthetic data, where the quality of the reference measurement is unquestionable.

2.2.2 The Necessity of a Validation Process

A lot of research has been contributed to the field of underwater localization in the past years. One of the driving reasons for this high interest is the necessity for new algorithms which can support the multitude of new scenarios for which underwater vehicles are used. The problem of underwater localization has a high order of complexity since a number of adverse conditions do apply:

- 6 DOF problem domain

The problem of localization underwater has to be inherently handled in all degrees of freedom, as an underwater vehicle can (and will) be moving in all these degrees of freedom. Even special cases of bottom vehicles (crawlers) or surface vehicles (boats) will be subject to rough terrain or waves respectively. For all vehicles

navigating in the water column, this is further aggravated by the inability to keep a position without active control.

- No easy access to reference localization measurements

In contrast to terrestrial applications there is no convenient reference localization information available. The signals of GPS do not penetrate the water far enough to be usable underwater (only a few centimeters), and long antennae or surface buoys are impracticable or even unusable due to currents and long distances from the surface (e.g. in the case of deep-ocean exploration).

- Reduced communication capabilities

Since water is a very good absorber in the EM-Spectrum, communication using radio is reduced to very short distances. While the modality of choice for communications is acoustics, its bandwidth and reliability is far inferior to terrestrial techniques (e.g. Iridium, WiFi).

- High effort for testing

The number of available large controlled-condition testing facilities for underwater vehicles is severely limited. This makes the usage of the ocean or lakes necessary for testing, which come with a huge logistic overhead and offer very limited control over the environmental conditions (e.g. salinity, turbidity, currents). The most interesting applications for underwater localization take place in the ocean, the largest biosphere on earth.

While many advances in underwater localization were made, it is hard to compare different methods. One of the reasons for this is the lack of a suiting terminology for comparison, another the problem of acquiring reference measurements under realistic conditions, due to the reasons listed above. As a result it is the author's believe that good practice for development of new algorithms includes the following points:

- Comparison with reference measurement

The new localization algorithm (NLA) has to be compared with a reference localization (GSRL) in order to be characterized properly. This reference measurement should be as good as possible. However the capabilities of the test vehicle and the specifics of the target environment need to be taken into consideration.

2. METHODOLOGY

- In-system validation

Ideally the reference measurement is recorded simultaneously with the NLA measurement, in order to avoid bias in the data. This means that both algorithms run synchronized on the test vehicle.

- Validation datasets including real-world data

The validation datasets should not only consist of simulation data or controlled-environment (e.g. test-tank) data. Ideally validation data is also recorded in the environment the NLA is designed for.

- Definition and description of failure cases

Depending on the sensors of the NLA, failure cases have to be considered. This may include environmental conditions (turbidity, inhomogeneous medium) as well as system-related conditions (available memory/processing power, power consumption).

The author is aware that not all of these conditions may be satisfiable for any given circumstance. Especially the availability of a good reference measurement for real-world environments is often limited. This is aggravated by the inherent recursiveness of the validation process: any reference measurement needs to be validated by another reference measurement, which makes it practically impossible to create a complete chain of validation. The only possible remedy to solve this problem is deviating from the necessity of real-world data for validation. For underwater localization this may mean using reference data from a test-tank with specialized short-range very-high accuracy localization systems (such as the VICON tracking system with underwater modification¹) or limitation to surface tests in open waters while using GPS as reference. A widely used practice is the validation using spot-measurements, e.g. using GPS localization at the start and end points of an underwater run while driving a closed path (39). All of these practices are non-ideal and provide a sparse validation, but suffice for the reference measurement validation. The reason for this is that the reference measurement should use an accomplished localization strategy. Accomplished in this context means a strategy which has been used by other groups with published success. Combining the knowledge of feasibility of a given GSRM with the sparse validation of a specific implementation results in sufficient dependability on the reference measurement.

¹<http://www.vicon.com/products/viconmx.html>

2.2.3 The Gold-Standard Method for Localization

The gold-standard method for localization (GSRL) of an underwater vehicle is a combination of infrastructure and vehicle localization systems on the basis of dead-reckoning. It uses the input from a number of sensors: IMU, DVL, DPS (digital pressure sensor) and LBL (long base-line beacons). All sensor data is fused by a KF (Kalman filter), yielding a 6 DOF localization measurement η . The different sensors all provide specific information for this purpose:

- IMU

The IMU provides the vehicle orientation η_2 and translational acceleration $\ddot{\eta}_1$. This data is high frequency (≈ 10 Hz), but the orientation (especially ψ) prone to drift and the translational accelerations suffer from random walk.
- DVL

The DVL provides vehicle speed over ground η_1 . This data is medium frequency (≈ 10 Hz) but dependent on external measurements, so prone to noise and drop-outs.
- DPS

The DPS provides distance to surface, i.e. the depth of the vehicle z . This data is medium frequency (≈ 10 Hz) and long-term stable.
- LBL

The LBL provides vehicle position η_1 data relative to one of the localization beacons. This data is low frequency (≈ 1 Hz) and low accuracy but does not suffer from long-term drift.

The quality of the localization is mostly dependent on data quality from the IMU, the DVL and the DPS, while the LBL sensor is used to assure long-term stability and cancel drift. Especially IMU sensors are available in highly different qualities, sizes and price ranges as well as availability. For example the JHUROV vehicle of the Johns Hopkins University uses an IMU developed by the US Navy, which is not available for non-military applications, but has very high levels of accuracy (57).

Most implementations of this localization technique are vehicle specific, there is no readily available open-source version which can be used out-of-the-box. This results

2. METHODOLOGY

in repeated implementation of algorithms and greatly reduces the comparability of individual vehicle's performance.

Examples for implementation of GSRL-flavours are the implementation on the DeapthX-Vehicle (19), where DPS, IMU and DVL sensors are used. The particular implementation with the vehicle's sensors yields a divergence rate of 0.5 % of distance traveled, although the method for determining this quantity is not given in their published work. The non-stabilized long-term error of 0.5 % fits with other publications, where similar accuracies are reported (8). A GSRL implementation including external references for long-term stability is described in (62). They use a high-quality DVL with 1.2 MHz combined with a 12 kHz LBL system to achieve a standard deviation of only 9 cm with an update rate of 4 Hz. The LBL provides position measurements every 6 seconds. This shows the effectiveness of the combination of a high-speed local sensor with a low-speed global sensor.

The implementation of the GSRL used for the work described in this paper follows a modular approach and is publicly available as open-source software using the ROCK-Toolkit (50) as middleware. The idea behind this it to enable easy adaptation to other vehicles or sensor configurations. A good example is the HROV-Project¹, in which parts of this implementation are used, even though the instrumentation of the HROV-vehicle differs considerably.

2.2.4 Validation Process and Performance Metrics

The steps to a successful validation of a new localization algorithm are:

- Selection and implementation of GSRL suitable for vehicle
- Sparse validation of GSRL implementation
- Implementation of NLA
- Recording of validation data
- Computation of resulting characteristics for NLA

In the following sections this process will be described in detail.

¹<http://robotik.dfki-bremen.de/de/forschung/projekte/hrov-arch.html>

2.2.4.1 Selection and Implementation of GSRL

The GSRL and its implementation used in this thesis are described in detail in chapter 4.

2.2.4.2 Sparse Validation of GSRL Implementation

The sparse validation of the GSRL implementation is described in detail in chapter 5.

2.2.4.3 Implementation of NLA

The implementation of the NLA is the core of the development and validation process. The implementation must be able to run side-by-side with the GSRL in order for the validation to work properly. The NLA and its implementation used in this thesis are described in detail in chapter 4.

2.2.4.4 Recording of Validation Data

The recording of validation data is described in detail in chapter 5.

2.2.4.5 Performance Metrics

The most important aspect is the computation of the characteristics of the NLA. The following quantities will be used for this purpose throughout the rest of this work:

The vehicle pose

$$\eta = [x, y, z, \phi, \theta, \psi] \quad (2.1)$$

in 6 DOFs, consisting of the vehicle position

$$\eta_1 = [x, y, z] \quad (2.2)$$

and its Euler orientation

$$\eta_2 = [\phi, \theta, \psi] \quad (2.3)$$

as described in section 2.2.1. The position at a given point in time is denoted by the use of $\eta_1(i)$ where $i = 0$ is the initial position and $i = n$ is the final position of an experiment. The position difference between a frame and its predecessor is defined as

$$\eta_{1\Delta}(i) = \eta_1(i) - \eta_1(i - 1) \quad (2.4)$$

2. METHODOLOGY

The distance traveled as measured by the algorithm can then be defined as

$$|\eta| = \sum_{i=0}^n |\eta_{1\Delta}(i)| \quad (2.5)$$

The deviation d is defined as

$$d = |\eta_1(n)_{\text{GSRL}} - \eta_1(n)_{\text{NLA}}| \quad (2.6)$$

and denotes the distance between the estimates of the GSRL and the NLA at the end of a test run.

A similar quantity is the sparse deviation d_s which is defined as

$$d_s = |\eta_1(0) - \eta_1(n)| \quad (2.7)$$

and denotes the distance between the position estimates of the start- and end of a test run for closed trajectories.

The in-track deviation $d_t(i)$ is defined as

$$d_t(i) = |\eta_1(i)_{\text{GSRL}} - \eta_1(i)_{\text{NLA}}| \quad (2.8)$$

with the mean μ

$$\mu = \sum_{i=0}^n d_t(i) n^{-1} \quad (2.9)$$

and the standard deviation σ

$$\sigma = \sqrt{\sum_{i=0}^n (d_t(i) - \mu)^2 n^{-1}} \quad (2.10)$$

can be used to judge the error characteristics between two estimated trajectories.

The relative deviation d_r

$$d_r = \frac{d}{|\eta|} \quad (2.11)$$

and the sparse relative deviation $d_{r,s}$

$$d_{r,s} = \frac{d_s}{|\eta|} \quad (2.12)$$

are used for distance-normalized comparisons.

It is important to note that all quantities comparing the GSRL and the NLA have to be corrected for the GSRL uncertainty. This is necessary, since the GSRL measurement

has an uncertainty as well, which needs to be taken into account. This has the inherent disadvantage that a NLA cannot be better than the GSRL in this framework. The correction for the deviation is simply done by adding the GSRL’s deviation, resulting in the corrected deviation d_c :

$$d_c = |\eta_1(n)_{\text{GSRL}} - \eta_1(n)_{\text{NLA}}| + d(\text{GSRL}) \quad (2.13)$$

This yields an upper bound for the NLA’s deviation. It is important to note that the tested algorithm (NLA) might perform significantly better than the reference algorithm (GSRL) without this being visible using this quantity. Nevertheless using this more conservative value, overconfidence into a new algorithm can be prevented.

For the graph-SLAM approach used in this work some additional graph-related quantifications are of relevance. For the graph $G = (V, E)$ with the nodes V and the edges E the size of the resulting graph defined as number of nodes $|V|$. The number of loop closures in the graph is the number of additional edges beyond a simple linear graph $L = |E| - (|V| - 1)$. The length of a path $p(V_i, V_j)$ between two arbitrary nodes i, j can be obtained by a breadth-first search on the graph, yielding $|p(V_i, V_j)|$. This is especially interesting for the nodes $i = 1, j = n$, which represents the length of the path from start node to the final node. The longest path in the graph is calculated by a complete breadth-first search from the start node.

2.3 Real-Time Constraints

One of the specific problems of underwater localization is the fact that since the vehicle is moving freely through the water column, it is usually not possible to pause motion in order to execute time-consuming computations. This inherently means, that if an algorithm is to be used on a live vehicle, it has to be real-time capable. This means that it needs to have a high update rate. A number of factors contribute to an algorithm’s update rate: sensor rates, processing demand, available processing power, parallelization, timing constraints. When designing an algorithm, a target update rate is required in order to select the participating components respectively. For the purpose of this work “real-time” means an update rate of approx. 10 Hz.

2. METHODOLOGY

3

Design of a Versatile AUV for High Precision Visual Mapping and Algorithm Evaluation

3.1 Introduction

In this chapter the work on the design, integration and field-testing of an underwater vehicle for visual mapping is described. The vehicle was specifically designed as a scientific AUV for visual mapping, incorporating high-end instrumentation and sensors to allow research in this area with the best technology can offer today. The complete process of development is described, starting with the decision to build a new AUV, then the design criteria, application for funding, integration, operation and ending with the evaluation of the vehicle after two years of operation.

Visual mapping is becoming of increasing interest in the underwater society. A precise and fast means of creating visual maps has a number of important applications, e.g. visual inspection of underwater structures, resource exploration or underwater archeology (45), (63), (23), (28). Besides the algorithms and software needed for such mapping tasks, there are a number of requirements for the vehicle actually performing such mappings.

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

3.1.1 Decision to Build a New AUV

It is a major decision for a research group to begin the lengthy endeavor of building an AUV. It is usually only considered, when it becomes apparent, that no other vehicle suiting the needs for the intended research is available on the market. This may have a number of reasons e.g. instrumentation, software-access to control/sensors, mass, size, endurance, depth-rating and export-restrictions. There might also be the issue of cost, but when the manpower necessary to completely build a new AUV from the scratch is factored-in this usually will become an argument *for* buying an existing vehicle.

At the beginning of this thesis the AUV “AVALON” (Aquatic Vehicle for Autonomous Learning, Operation and Navigation, shown in figure 3.1) had already been built by the author and used for preliminary experiments. Envisioned as a low-cost (student) research vehicle, its design criteria were much simpler. AVALON consists of two pressure hulls connected by superstructure, with two thrusters (one diving and one horizontal) in between. The two driving thrusters are mounted besides the vehicle in the center area. This concept allows the thrusters to apply their force near the center of gravity (COG), which results in less disturbances in attitude when diving/moving horizontally. Two more thrusters were mounted at the rear of the vehicle for diving and horizontal movement. The maneuverability of AVALON was excellent, the concept of six thrusters for actuation of five degrees of freedom (DOF) confirmed as benevolent. The other DOF, roll was kept stable by a low COG. This was achieved by a very simple means: the lower half of each pressure hull is filled with the batteries. Since the batteries weigh a lot more as the electronics mounted on top of them, the COG is kept low without the need for additional lead or a keel, and roll movement was limited. A drawback of using two pressure hulls is the need to connect any electronics between the hulls with underwater connectors which tends to increase the amount of plugs and cabling significantly.

Since AVALON had seen more than 100 hours of active operation in a number of environments (starting in swimming pools indoors, over lakes up to the open ocean) the flaws and problems of its design were well documented. While some of the graver flaws could be directly addressed on AVALON, others were too intrinsic or would have resulted in major modifications:

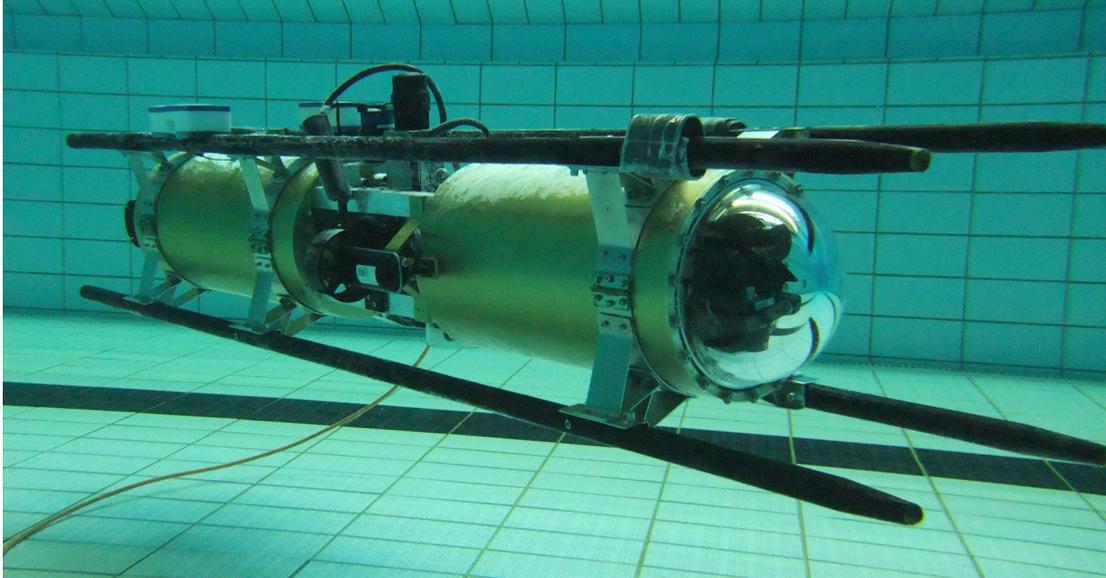


Figure 3.1: The AUV “AVALON”.

- **Mechanical**
AVALON’s mechanical design was very simple, which led to a number of problems with maintenance and handling on the long run.
- **Control**
The vehicle’s thrusters were six Seabotix ROV thrusters with only PWM control and no sensor feedback, which limited precision control and repeatability.
- **Instrumentation**
The vehicle was only equipped with most basic sensors. An analogue pressure sensor, altimeter and AHRS limited navigation capabilities.
- **Cameras**
AVALON had only two cameras, one observing the sea-floor, the other looking forwards. Due to the vehicle design the maximally possible stereo baseline was 13 cm, which would have limited depth resolution significantly. Additionally the cameras were placed behind a large curved dome at the front of the vehicle, which led to significant image distortion.
- **Handling**
The battery only lasted about four hours and the vehicle had to be opened for

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

recharge. The length of the vehicle (1.8 m) limited its maneuverability in confined areas.

Other aspects of the vehicle had proven sensible and usable:

- Operating Depth

AVALON was designed for depths up to 150 m. This enabled the usage of relatively simple and cheap components while at the same time making the vehicle fit for all applications in local waters. The low depth rating also directly resulted in a relatively light vehicle (85 kg) which facilitated handling.

- Software Framework

The ROCK framework used on AVALON was still in early development, but had proven a good basis on which an autonomous vehicle could operate.

- Control Design

The ability of the vehicle to hover and freely select its forward speed, as well as being able to move sideways were very helpful in data acquisition for algorithm development.

- PC-Based architecture

The two control computers were PC-based (Mini-ITX Intel Core2Duo). While not very power efficient (hotel load: 80 W) this enabled the usage of non-specialized software and fast software development.

One of the major problems was that since the available instrumentation was very limited, no localization was possible using established methods. While a simple visual algorithm could be implemented using the bottom camera and an optical-flow approach, a major problem remained the verification of its correctness. Further, this solution was based on a monocular camera, and thus suffered from scale ambiguity.

When it became clear that AVALON would not be fit for the purpose of this thesis, a market analysis was conducted to get an overview of possibilities. Unfortunately, no system was found, which suited the specific needs (see section 3.2), not even partially. All systems were either too heavy for operative handling, too expensive or too mildly instrumented. The ability to access all sensors and directly change vehicle behavior was also something missing on most commercial vehicles. This led to the decision

that a completely new vehicle was necessary. This vehicle would be built on the basis of AVALON and the experiences gained there. The resulting process will be described in the following sections.

3.1.2 External Help

The vehicle was designed within the scope of this thesis and the publicly funded CUS-LAM project. While most of the work on the idea, design, specifications, instrumentation and application has been done by the author, he had help in other areas from colleagues, especially in areas which surpassed the scope of his skills. For the mechanical design Jens Hilljegerdes created all engineering drawings which were sent to the manufacturers. He also integrated the tiltable camera head. During the integration of the vehicle Marius Wirtz helped in the many “small” things: soldering, fixtures, assembly. The vehicle control software was mostly developed by Christopher Gaudig who wrote the code for the individual behaviors (e.g. hovering, obstacle avoidance). Sankaranarayanan Natarajan implemented the parameter identification algorithm for the mathematical vehicle model as well as the trajectory follower. Leif Christensen did the FPGA implementation for the thruster control. Patrick Paranhos and Javier Carrio implemented the Kalman filter for sensor fusion (“pose-estimator”, see section 3.3.8).

3.2 Design Criteria

The design criteria for the new AUV can be roughly separated into mechanical criteria, describing the dimensions, actuation system and overall appearance, and into the sensor requirements, dictating the available sensory equipment of the vehicle. Both constraints will be described in the following two sections.

3.2.1 Mechanical Requirements

There are a number of design criteria which were considered high priority primary parameters. The vehicle should be small, ideally not exceeding 60 kg for ease of deployment and handling. Diving depth should be at least 150 m, in order to retain the possibility of surveying near-shore continental shelves. The speed over ground is required to be freely selectable (in reasonable ranges), in order to test algorithms at different

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

speeds. In order to operate within narrow constraints of structures, high maneuverability and rate of turn are necessary. By attachment of a fiber-optic cable, the experiments should be supervizable while keeping the diameter of the cable as low as possible to reduce the induced movement impedance. Parameters usually of high importance (long battery life, high speed, low hydrodynamic drag) only are of secondary interest, since they would interfere with the primary parameters in a negative way, or increase the cost of the vehicle.

3.2.2 Sensor/Instrument Requirements

One basic problem in development of new navigation and mapping algorithms is measurement of their accurateness and robustness. In order to do so, ideally a ground-truth measurement should be available to which the new estimate can be compared. This usually is not the case in underwater environments, since highly precise and frequent absolute position measurements are hard to achieve. This approach aims to use state-of-the-art sensors and technologies to get the best position measurement possible with “traditional” methods, meaning a combination of external reference measurements (LBL, USBL), speed measurements (DVL) and inertial measurements (IMU, FOG). Since such systems have been widely used in the underwater community, their precision and performance is well known and documented. These measurements will be used as gold-standard to compare newly developed, visual algorithms against. Of course, this means that a complete set of classical navigation instrumentation is required on the vehicle besides the camera system.

Since the camera system will be the main payload sensor on the system, a number of requirements exist for this sensor. The camera should be a color stereo camera with at least 20 cm baseline between the lenses. One of the problems with stereo cameras is the rigidity of the rig against external deformation - in order to avoid such problems special care has to be taken to connect the cameras as rigidly as possible. The field of view (FOV) of the cameras should be as large as possible, in order to maximize image overlap both between stereo pairs and between consecutive images. The cameras should be ground tracking, their angle to ground be freely selectable between 0° and 45° . The cameras should have high physical resolution in order to get high quality visual data as input for the algorithms. Digital cameras are preferable to their analogue counterparts to minimize image noise. The achievable frame rates should ideally be video frame rates

(25 Hz), with a lower limit of 15 Hz. The usage of cameras equipped with highly sensitive CCDs has the advantage of reducing the illumination requirements. Using lenses with a large aperture has the same positive effect, and should be used in combination with the former. Illumination should be as uniform as possible in the entire field of view. The last parameter for the camera system is the depth of field of the lenses, which should be as large as possible in order to avoid image blur.

Processing of high-resolution camera data requires a lot of processing power. Besides a control and guidance computer system a second system is required which solely handles the image processing tasks. Separation between the two systems is important, since overload caused by image processing tasks could adversely affect vehicle control, which is unacceptable. The aim is to get as much processing power into the system while maintaining power consumption low. Since it has been shown that stereo processing can be done on graphics cards (GPUs) very well ((65), (58)) an inclusion of such a device on the AUV would allow interesting possibilities. The cost in power consumption however is relatively high for such devices.

3.2.3 Vehicle Concept

The main purpose for the design of the AUV, is to build a multi functional vehicle in shape, mobility and integrated sensors. The design has also to combine the option of a streamlined shape with the option to install a stereo vision system that uses a defined distance between the cameras. The tube shape is as simple as well as effective design for AUVs. Unfortunately the integration of a stereo vision system into such a cigar-shaped vehicle is difficult: either the baseline of the stereo system is very small, or the tube diameter grows very large. This makes this design ineffective. A frame-based vehicle seems more appropriate for incorporation of a stereo camera. Open frame based AUVs often include separate water-proof containers which are connected with under-water connectors. Because of the effort and space of the containers this design is difficult to install in a small size system with a large number of equipment. Also a frame construction in the shape of a tube (often realized with additional covers for reduction of drag) increases the ratio of system volume to usable space, and thus weight, which is not desirable for this application.

It was decided that a hovering AUV with five active degrees of freedom is needed. The remaining DOF (roll) is to be passively stable. To reach this the basic design

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION



Figure 3.2: Parallel setup without peripheral equipment.

of AVALON was taken, but not with the two hulls aligned to form a cigar-shaped structure, but besides each other in a more catamaran-type configuration (see figure 3.2). This has a number of advantages: the stereo baseline can be greater than the hull diameter, because it is now only limited by distance of the hulls. The center of gravity is inside the vehicle, but since there is open water in between the hulls the diving thrusters can be mounted between the hulls, achieving high effectiveness. Additionally to the diving thrusters there is space between the hulls to accommodate additional sensors (e.g. the DVL). The whole vehicle is kept very compact, which reduces its tendency to pitch or roll. Especially the latter is of high importance, since it is the passive degree of freedom. The compactness also improves maneuverability, which will be helpful in confined spaces. A disadvantage is the higher water resistance and thus reduced endurance. Since these parameters were of secondary nature, this was deemed acceptable. The basic idea of the two hulls containing the batteries in the lower half, and the other electronics in the upper half was kept, in order to keep the COG low. The two hulls should be connected by dry tubes, reducing the amount of underwater connectors by wiring any connection between the two hulls through these dry tubes.

Instead of mounting the cameras behind acrylic domes it was decided to put the cameras onto a tilt unit, which can be tilted 180 degrees around the pitch axis. This

has two advantages: the view port of the cameras can be flat (which facilitates camera calibration) and made of standard glass (which improves the pressure rating), and the camera viewing angle can be selected very easily. The price for this setup is a more complex head design. Similarly to the main hulls the electronics of the head is connected dryly to the main hull, making the vehicle one big pressure hull. In order to protect the system from water in case of a leak, the two heads and the two hulls are sealed with low-pressure sealants from each other.

These thoughts result in the following basic specifications for the AUV:

- **Navigation**

- LBL/USLB tracking system transponders integrated
- DVL
- IMU
- FOG
- Pressure sensor
- HD Stereo Camera

- **Communication**

- Fibreoptic cable link
- Telemetry modem

- **Dimensions**

- 700x600x300 mm outer dimensions
- 85 kg weight in air

- **Instruments**

- 2 Embedded PC systems
- 1.6 kWh Lithium-Ion Battery @ 29.6 V
- approx. 5000 lm worth of light
- 5 brushless thrusters, 2.5 kg bollard thrust @ 150 W

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

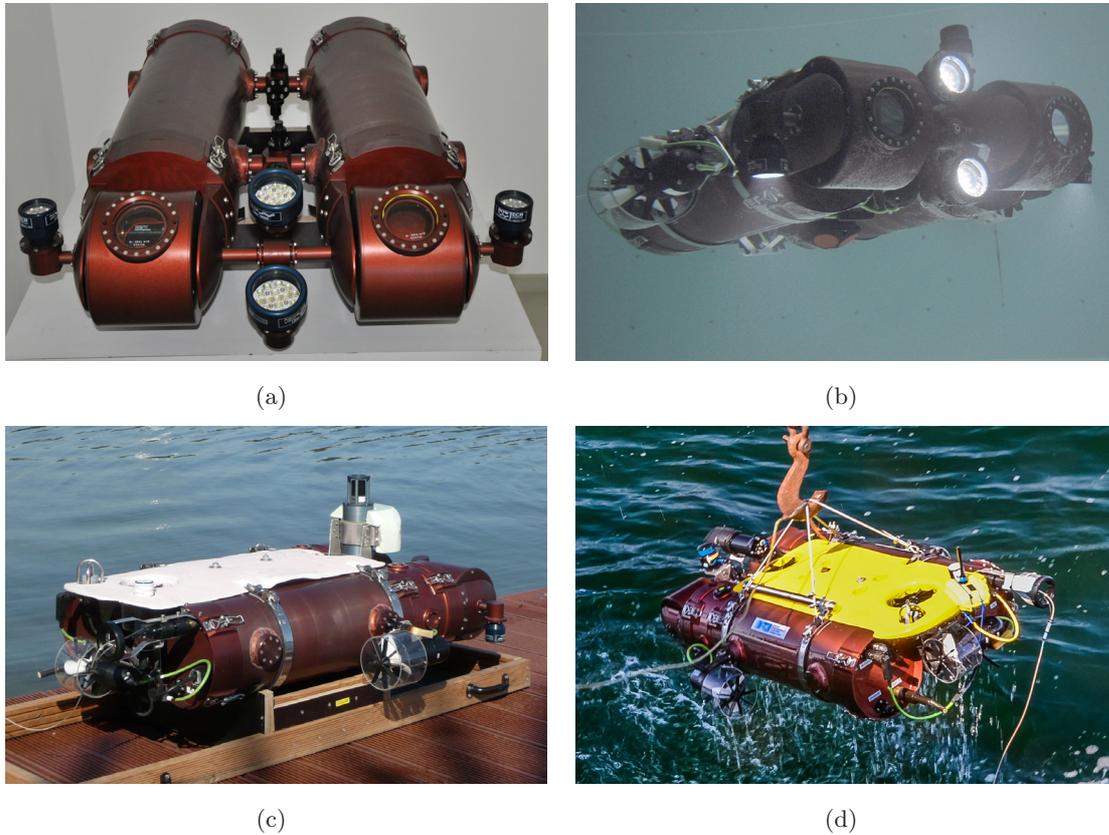


Figure 3.3: The AUV DAGON evolving: pressure hull testing (a); basic indoor testing (b); outdoor testing (c); current state (2013). (d).

3.3 System Description

Mid 2009 the cost for such a vehicle was estimated and a project proposal was drawn and sent to a funding agency (BMW, German Ministry of Economics), which granted funding for the CUSLAM project under the grant No. 03SX290. The CUSLAM project provided material as well as personal costs. After the project's kick-off in 2009 the final work on the design of the vehicle and its integration could start. The result was the AUV "DAGON"¹, which will be described in this section. An overview of DAGON's instrumentation can be seen in table 3.1. All of the required sensors could be integrated into the system. DAGON's evolution over the past years is shown in figure 3.3.

¹DAGON is not an acronym, but a reference to an aquatic deity from H.P. Lovecraft's Cthulhu mythos

3.3 System Description

Table 3.1: List of sensors and instruments of the AUV.

Instrument	Property	Rate	Precision	Range
XSens MTi AHRS	Attitude (R/P/Y)	120 Hz	0.5 ° (R/P) 1 ° (Y)	360 °
KVH DSP-3000 single axis FOG	Yaw rate	100 Hz	1-6 °/h ⁻¹	±375 °s ⁻¹
Desert Star SSP-1 pressure sensor	Depth	0.25 Hz to 16 Hz	0.1 % RMS	0 m to 344 m
Desert Star SAM-1 acoustic modem	Telemetry	23 bit s ⁻¹	-	250 m to 1000 m
Desert Star VLT-3 LBL transponder	XYZ position	0.2 Hz to 2 Hz	±0.15 m	2000 m
Teledyne RDI Explorer DVL	Speed over ground	12 Hz	±0.007-0.03 m s ⁻¹	0.3 m to 80 m
Micron DST scanning sonar	Distance	0.5 Hz ^a	-	2 m to 75 m
Micron USBL transponder	Range/Bearing ^b	0.1 Hz to 2 Hz	±0.2 m, ±3 °	150 m to 500 m
2 Bowtech LED3200	Illumination	22 kHz PWM	255 steps dimmable	-
2 AVT GE1900C GigE-cameras	Image	0-30 FPS	Full-HD (1920x1080)	-
1 AVT GC1380HC GigE-camera	Image	0-30 FPS	HD (1380x1024)	-

^aFor 360 ° scan

^bRelative to receiver

3.3.1 Pressure Hull

The AUV design consists of two main tubes with equal supports for the rear and front cap. This construction enables a various number of combinations using different caps. The tube itself is a welding construction with bonded rings on the ends for the cap locking device. The material in use is depended on corrosion (ALMg4,5Mn) and availability (ALMg3). The coating is a red colored hard-coating which gives a much

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

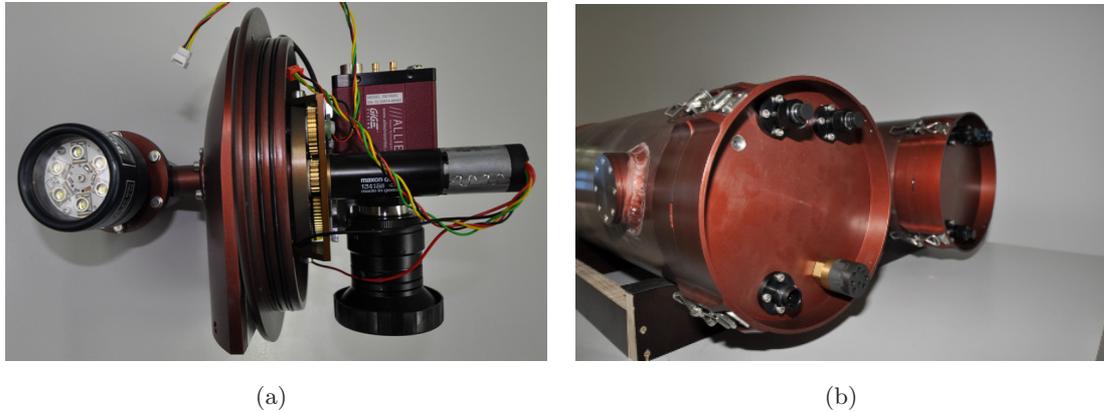


Figure 3.4: Head mechanics with camera and fixture for power electronics (a); Rear caps and quick fasteners (b).

better control to the fit tolerances to other techniques like anodization. Attached to the front of the two main hulls is the stereo head. The main setup for the stereo based system can be seen in figure 3.3(a). For an easy access and for maintenance reasons the caps are fixed with quick-release clamps.

For the primary stereo vision setup the front cap is equipped with a 180 degree turnable tilt unit. The two main tubes are connected with side connectors parallel to make the two front cameras aligned on a turnable horizontal axis. To realize a fixed position of the two cameras in the offset angle they are connected with a horizontal tube construction. This tube is also used to fix two of the four LED lamps, so that the lamps are always aligned with the cameras. On the left and right side there are two additional LED headlamps in a fixed down position.

Each of the head tilt units is independently driven by a gear motor with a gear ratio of 1/1014. Combined with a gear ratio of 1/5 installed on the head main axis the available torque is limited by the gear shaft up to 10 N m. The motor requires 25 s for a complete 180 degrees tilt. The head mechanic is shown in figure 3.4(a).

3.3.2 Battery

The battery used for DAGON is a lithium-ion battery system. It consists of two individual serialized blocks with 14.7 V and 50 A h, resulting in a total capacity of about 1.5 kW and an endurance of about 8 h to 10 h. The battery management system supervising charge and discharge of the battery is capable of showing the current rest

capacity and can be used to determine how much time can be spent before re-charging. A quick-charge is possible in two hours, although this is only possible when the vehicle is not being operated at the same time due to temperature limitations. The battery system has proven very reliable, especially as compared to the battery system used on DAGON's predecessor AVALON, which had to be replaced twice already due to damaged cells.

3.3.3 Data Processing and Communications

The selection of the main processing components for a mobile robot is a critical design choice. The range of available components is huge, a number of criteria have to be evaluated for selection. The first criteria is COTS (commercial off the shelf) vs. custom design. While custom design will always be superior with regards to the specific needs of the target vehicle, its drawbacks are numerous: increased development time (and thus overall cost), limited availability, limited experience with system, limited knowledge of failure behavior. This has led to the decision to go for a COTS-based solution for the main processing, and only use custom design where it is necessary (e.g. the electronic fuse, see section 3.3.5). The next criterion is processing power vs. power consumption. Microcontroller based solutions have the lowest power consumption (e.g. RaspberryPI, 5 W including peripherals ¹), with the drawback that their processing power is severely limited and the programming can not be done in higher languages. FPGA (field programmable gate array) based solutions can be very computationally powerful while requiring reasonable amounts of power (e.g. Xilinx Virtex5 XC5LX330, 23 W (12)), but the drawback is the complexity of implementation (usually VHDL) and re-usability of pre-existing code. Embedded PCs are the next candidate group, which ranges from Low-Power (e.g. Intel Atom Z530, 2 W ²) to high power (e.g. Intel i7-4770S, 65 W ³) processors. The advantage of embedded PCs is their usage of the x86 instruction set, so programming on them is exactly the same as for a desktop PC. With the wide range of different CPUs available this solution was determined to be the most

¹<http://www.raspberrypi.org/faq#powerReqs>

²http://ark.intel.com/de/products/35463/Intel-Atom-Processor-Z530-512K-Cache-1_60-GHz-533-MHz-FSB

³http://ark.intel.com/de/products/75124/Intel-Core-i7-4770S-Processor-8M-Cache-up-to-3_90-GHz

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

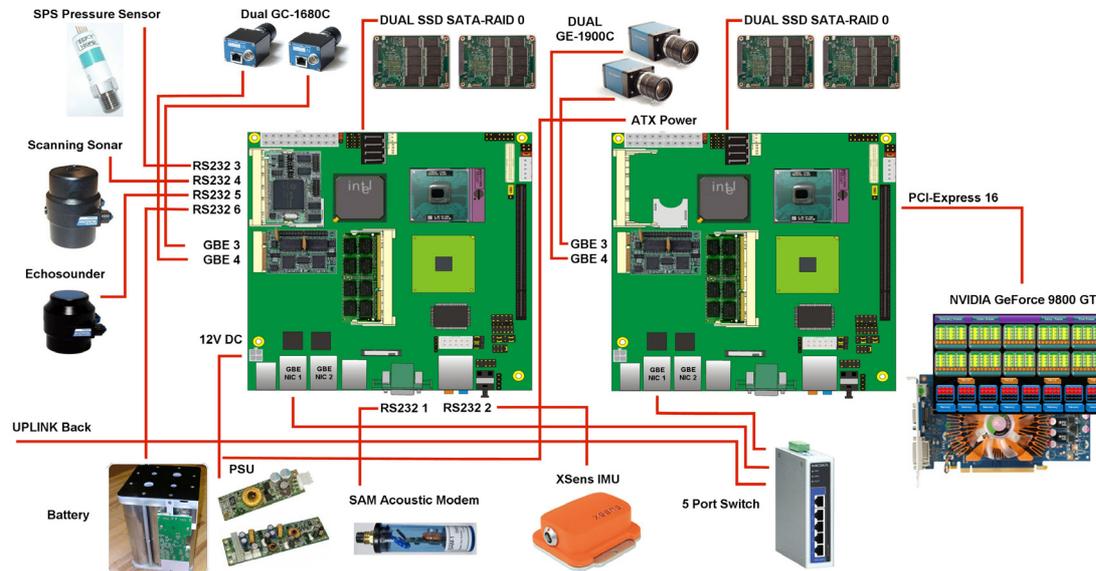


Figure 3.5: Data and power connections within the vehicle.

reasonable, while the usage of FPGAs was kept in mind as possible co-processor for the future.

For DAGON two embedded PCs were selected, one located in each pressure hull. Both PCs have to deal with data acquisition from the sensors, depending on their location in the hull as well as communication with the actuators. The right PC is used for vehicle control and the computation of the reference localization, while the left PC is executing the visual SLAM. Initially both PC were equipped with LV67B mainboards by Commell and Intel Core2-Duo P9600 CPUs with 25 W TDP (thermal design power)¹. Later the left PC was replaced by a IEI KINO-HM551 and an Intel Core-i7 620M CPU with 35 W TDP². This change was necessary to accommodate a GPU (graphics processing unit) as co-processor as described in section 4.2.4.2. Both PCs are equipped with solid-state hard-drives as system drives with an additional mechanical hard-drive for data logging. The left PC has a RAID-0 disk array for storing camera images coming in at high data rates (up to 120 MB per second). Both PCs are connected over a Gigabit-Ethernet link, which is also connected to the surface cable/WiFi.

The available data link options are:

¹http://ark.intel.com/de/products/37266/Intel-Core2-Duo-Processor-P9600-6M-Cache-2_66-GHz-1066-MHz-FSB

²http://ark.intel.com/de/products/43560/Intel-Core-i7-620M-Processor-4M-Cache-2_66-GHz

- 54 MBps WiFi link (surface)
- 1 GBps copper cable (indoor tests)
- 1 GBps fiber-optic cable
- 15 Bps acoustic modem

The WiFi link is only available on the surface and is used for setup and starting of autonomous missions as well as vehicle recovery after such a mission. The copper cable has a length of 50 m (can be extended to 80 m) and is used for indoor testing. The main communications link is the fiber-optic cable (see section 3.3.4). The acoustic modem is used for low bandwidth status updates during the submerged phase of autonomous missions, and can also be used to send a mission abort signal (which makes the vehicle come back to the surface).

3.3.4 Fiber-Optic Cable

DAGON has a Seacon fiber-optic cable port. This allows the connection of a single strand amide-reinforced fiber cable, which can be used to monitor the AUV during missions. While this violates the idea of an autonomous underwater vehicle, effectively demoting DAGON to a ROV with its own power supply, this option is invaluable. Since it is impossible to transfer broad-band data through the water-column, without a hard link the AUV and its behavior can only be observed in retrospect, by analyzing the recorded data. Debugging like this can be very tedious, especially during the early development stages, where many of the basic behaviors still have to be tested and tuned. A traditional copper data cable suffers from low transmission ranges or low data rates, and heavily impacts vehicle control, since it either is heavy in water, or made neutrally buoyant with floaters, increasing its diameter and thus drag. With a diameter of only 1.6 mm, neutral buoyancy in fresh water and a tensile strength of 1200 N, the fiber used is very slender. Still it can support data rates of full-duplex Gigabit Ethernet over a distance of 20 km using CWDM technology¹.

The 500 m of fiber available have been used excessively during field trials. The typical mode of use was the following: Even though a data line was available all processes and computations were run on the vehicle PCs. The only processes allowed on surface

¹In CWDM the RX and TX data channels operate on different frequencies of light in a single fiber

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

PCs were for monitoring purposes (e.g. camera image viewer, behavior inspector). This ensures a clean transition to fully autonomous modes without cable. During the final tests it often occurred that DAGON was executing a mission while the operations team was simply observing to make sure everything was working well. The ability to quickly modify software or adapt it to a change in the environment has been invaluable.

The downside of the use of a fiber cable is its fragility and its cost. In contrast to copper cable great care has to be taken to not over-bend it. The cable used was exhibiting a tendency of forming loops, which according to the manufacturer originated in the manufacturing process and could not be remedied. If a loop was put under strain, it could tighten to a degree where the data transmission was interrupted. Luckily the cable was not harmed in any of these occurrences, but this experience will influence the choice of future cables. A second issue is the cost. The components necessary for operation (bulkhead connector, cable connector, cable, reel with rotary joint, converters) do not come cheaply, and were complicated to procure in Europe (e.g. the cable had to be sent to the connector manufacturer in the US for cable molding).

3.3.5 Electronic Fuse

During the first year DAGON had to be frequently opened because of blown fuses. For security reasons all components are individually equipped with a one-way fuse. Especially the fuses of the thruster control boards tended to blow on a regular basis (due to over-current or jamming). Since opening of the vehicle is problematic, especially during field-tests, a solution for this was necessary. The DFKI-RIC's electronics department created an electronic fuse board. It has five individual power ports, each allowing up to 36 V with 10 A. A micro-controller monitors the output ports and deactivates them using a solid-state-relay if over-current is detected. The permitted current can be selected via a serial link. While this electronic fuse does not offer the same security as a mechanical fuse, it has proven a very good middle way, and performed without any problems.

3.3.6 Thrusters

Other than originally planned only five thrusters were built into the vehicle. The second lateral thruster was omitted. Its placement had been awkward below the main housing and initial experiments showed that while it provided extra stability when

moving sideways, it was not absolutely necessary to have this sixth thruster. The five thrusters used for DAGON have been developed specifically for this AUV. They consist of a brushless DC-motor, equipped with hall-sensors in a custom housing. The motor control electronics are built into the pressure housing as well, reducing the external cabling significantly: only 5 leads are required, two for DC power and three for CAN communications. The FPGA-based motor control electronics (called BLDC V1.3, developed at the DFKI-RIC) used the hall sensor information to accurately commute the brush-less motor. This allows extremely accurate control of the RPM even at low speeds (starting at 10 RPM), as opposed to sensor-less approaches, where RPM control is only possible at higher speeds. As a result the AUV is able to very accurately apply thrust – important for a hovering vehicle.

The downside of having such sophisticated thrusters was the high number of problems they were experiencing during the last two years. Since they were not thoroughly tested before integrated into the vehicle (due to time constraints), many initial quirks were still present. This resulted in a number of experiments being aborted due to thruster malfunction. During the outdoor tests in mid-2012, not a single thruster failed. This is regarded as a tentative signal, that now most of the problems have been identified and fixed. Still a more thorough testing before integrating them into an operational vehicle would be recommended for the future.

3.3.7 Vision System

As stated above, the stereo camera system is considered the main sensor system. The selected cameras, two Prosilica GE1900C Gigabit-Ethernet cameras are extremely sophisticated sensor systems. Equipped with a Kodak KAI-2093 1" CCD sensor with Full-HD resolution (1920x1080 pixels) and a quantum efficiency of more than 30 %, they offer crisp, low noise color images. The camera can record as many as 30 frames per second at full resolution, which is even beyond video frame-rates. Together with an Lensagon 8 mm 1" lens with an f-number of 1.4, the camera becomes a great instrument to visualize underwater scenery. The lens offers a diagonal FOV of 101° in air, which will translate into roughly 67° in water (the system uses a straight view-port). The selected baseline of the stereo camera system is 30 cm, which results in a stereo overlap of 92 % at three meters viewing distance from the sea-floor. At this distance the cameras have a single image swath of 4 m, which translates to a resolution of 2 mm

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

per pixel - an excellent value for mapping applications. Together with the powerful LED-based illumination system this camera system can be considered one of the most sophisticated setups in AUVs today.

Since the two cameras are to be used as stereo camera system, the timing during image acquisition is crucial (since DAGON is a moving vehicle, images captured at different times do not geometrically obey epipolar geometry). In order to avoid software-triggering, a hardware triggering mechanism was implemented. For this purpose, the left camera works as master camera, having a freely selectable FPS and exposure. When starting image acquisition, it changes the electric trigger signal to “high” state on the right camera’s trigger input, which in turn makes the right camera expose as well. At the end of the light exposition, the trigger signal is changed back to “low” again. This means that whenever the left camera is exposing, the right camera is exposing as well. Signal run-times have been compensated according to the manufacturer’s manual. This process results in precisely timed, equally exposed image pairs. As an additional benefit, it allows the left camera to operate in automatic exposure mode (for changing lighting conditions) with the right camera automatically adapting to the same exposure value.

The two cameras are mounted inside independently rotating tilting units. The sensors attached to each of the units (absolute encoders and incremental encoders) were supposed to assure knowledge of the current tilt position, and thus a correction of the calibration parameters (see section 5.2.3 for details). Unfortunately this did not work as designed, since the tilting mechanics had too much gear play (2-3°), making a re-calibration of the cameras necessary after each change in camera tilt.

3.3.8 Reference Localization System

The AUV is equipped with two absolute position measurement systems: a reverse-LBL and an USBL. The reverse-LBL allows the system to measure its own position relative to a grid of four pre-installed transponders in an area of about 500x500 m. This position measurement is used together with the DVL and the AHRS/FOG for ground-truth measurements as described above. Because of weight restrictions the vehicle LBL transponder was integrated into the main pressure hull of the AUV as opposed to its external fixture. Only the transducer and the pressure sensor are mounted in the water directly. The USBL solution is not meant for usage for vehicle navigation, but in order

to track the vehicle during autonomous surveys from a boat. The USBL transponder from Tritech is so small as not to impede the vehicle. Further detail on the reference localization system and its software implementation is given in chapter 4.1.

3.3.9 Basic Controllers

DAGON has a number of simple controllers, which are essential for most work with the vehicle. Most basic is the thruster controller. It allows the selection of a RPM for each thruster, which is then maintained by the electronic control board inside each individual thruster. The input for this thruster controller can come from a number of sources, depending on the mission profile. The most basic control path is the completely manual control, where a joystick controls all thruster movements. The joystick axes are linked with the thrusters by a simple thruster control matrix.

For additional controllers a simple PID-controller was implemented and parameterized for the specific task. It is used for the following controllers:

- *depth/altitude controller*

Using input from the pressure sensor (depth control) or the DVL (altitude control) the vehicle's position in the heave direction is maintained at the selected position.

- *pitch controller*

Using the two diving thrusters differentially, the pitch can be actively controlled in the range of $\pm 45^\circ$, using the AHRS as sensor.

- *heading controller*

The vehicle yaw is controlled using the orientation-estimator as input and applying thrust on the rear thruster to turn the vehicle

3.3.10 Station-Keeping

Station keeping is a very useful behavior, especially during field-testing when an external current was present. The underlying controller tries to maintain zero velocity on the x/y axes as well as the current heading. The DVL is used as input. While this does not necessarily result in absolute position keeping, it performs rather well. One of the problems observed during the sea-trials in the Baltic Sea was that since DAGON has only one thruster for lateral movement, it is sometimes not strong enough for station

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

keeping when a cross-current is present. This could be easily remedied by monitoring the direction of the current, and rotating the vehicle so it can use its surge thrusters to counter the current. This is still under development.

3.3.11 Trajectory-Follower

In order to survey or map pre-defined areas, a typical AUV application is the automatic and precise execution of trajectories. A trajectory is defined by a set of vertices creating the support polygon, which are interpolated by a C2-steady curve (10). The AUV controller then tries to follow that curve as precisely as possible. For this to work properly, the curve interpolation parameters have to be selected according to the AUV's capabilities with regard to turning radius and degrees of freedom. This is relatively easy for a hovering AUV like DAGON, since due to the fact that it can freely select its speed and point-turn, its minimum turning radius is zero. However since this would potentially produce unwanted behavior, the minimum curve radius at different traveling speeds was experimentally determined, and can now be used for parameterization of trajectory interpolation.

The vertices for the support polygon can either be in \mathbb{R}^2 or \mathbb{R}^3 . \mathbb{R}^2 vertices are used when the depth of the vehicle is to be determined by a different controller, e.g. terrain following or constant depth. \mathbb{R}^3 vertices can be used to navigate in known environments.

The implementation of the trajectory follower uses the input of a pose estimator (see section 3.3.8) to determine the vehicle's current position on the trajectory, and the vehicle motion commands and model together with input from the DVL to determine the vehicle's current speed and motion flexibility.

3.4 Operation

DAGON has been used in a number of environments, ranging from the test basin at the lab, in lakes, the open sea and under ice. A detailed description of the testing environments is given in chapter 5. These diverse environments each pose different operational circumstances and limitations, which had to be learned by the operations team. Experiences with different launch/recovery strategies, balancing/buoyancy tuning, external supervision and in-field maintenance will be given in this section.

3.4.1 Testing Environments

By mid 2012 the AUV DAGON has accumulated over 500 active hours in the water. Most of that time was spent in the underwater testbed at the DFKI-RIC, followed by a lake near the University of Bremen. Additionally two open water field-tests were conducted in the Baltic Sea, one in Kiel and the second near Rostock. Each of these diverse testing environments has its unique advantages and limitations. More detail on the testing environments and the experiments conducted there are given in chapter 5.

3.4.2 Field Testing

Due to the relatively small size of DAGON, launch and recovery proved to be both easy and challenging. A system of four fixtures (two each hull) where a harness was attached was used to lift the vehicle. This proved to be a safe system for putting the AUV into the test tank or launching it from the crane built into DFKI-RIC's own RIB. For the launch from the research vessel *Gadus* during the trials in the Baltic, the nylon rope harness was exchanged with one made of steel. Due to the high stiffness of this harness catching the AUV with a hook on *Gadus*' crane proved to be very feasible up to the sea states where DAGON can be operated in. For operation in shallow water environments like the gravel pit in Bremen, the maintenance sledge is used directly together two people with rubber trousers to deploy/recover the vehicle.

The main operation mode of DAGON is in an hybrid mode, where the fiber is attached to view the actual state and sensor output of the control software, even during run an autonomous mission. The ROCK framework allows for a seamless transition between direct controlled and autonomous operation. The system is started in the same way, weather it is tethered or operated via a WiFi link when on the surface before a full autonomous run. Via the underwater modem link basic commands can be given to the vehicle and receive status information. A TriTech MicroNav USBL system can be attached to DAGON giving a rough estimation about direction and distance during full autonomous operation. During the various field tests it was discovered, that DAGON could be operated safely in most conditions. The only problem encountered is the unreliability of the WiFi link due to reflections on a during higher sea states. Currently it is planned to adopt the dual-mode radio link used on the older AUV AVALON to get rid of these problems.

3. DESIGN OF A VERSATILE AUV FOR HIGH PRECISION VISUAL MAPPING AND ALGORITHM EVALUATION

The problem with a control software research vehicle is, that the internal electronics are changing a lot. Even during tests in the field there are sometimes changes, which is the source for numerous bugs (e.g. wiring problems). The layout of DAGON was done in such a way, that most internal parts are accessible for in-field maintenance. Both pressure hulls can be opened without special tools, the transport ledge is constructed in such a way, that the hulls open easily. The system is sealed by applying negative pressure. All tools and spare parts fit within several normal sized boxes, and DAGON can be transported with all necessary equipment within a van.

3.5 Evaluation

During the past three years the AUV DAGON has been developed, integrated, tested and used. Even though a lot of work could still be done to improve the vehicle, it can now be regarded as ready for further experimental use. It has proven its utility and versatility in a number of different environments and situations, hopefully remaining in service for a long time.

3.5.1 Future Work

There is a large number of ideas for further work. The following items are of high interest and will be hopefully implemented in the near future.

DAGON currently has a semi-open-frame hull. In order to improve its performance with regard to endurance and robustness to currents a streamlined outer hull made from glass fiber reinforced plastic is being planned. It is supposed to be easily detachable for maintenance and additionally provide some mechanical protection for the pressure hull and sensors.

4

Algorithm

In this chapter the underlying algorithms are described. It is separated into the description of the reference localization implementation on DAGON, and the details and implementation of the SURE-SLAM algorithm.

4.1 Reference Localization Implementation

The GSRL implementation on the AUV DAGON is shown in figure 4.1. At its core is a Kalman filter (called pose estimator) fusing the sensor data from all available sensor systems. In order to keep this Kalman filter simple, the orientation estimation is computed separately, fusing the information of a single-axis FOG (Fiber Optic Gyroscope) for yaw with the data from an AHRS (Attitude Heading Reference System). Due to cost restrictions a navigation-grade IMU was not feasible for the vehicle. Since strong magnetic disturbances were expected during operation (especially in the metal-walled test-tank) the magnetometers of the AHRS are only used once during initialization, resulting in the yaw portion of the orientation being measured only by integration of the FOG data during operation. This results in the following inputs: orientation η_2 from the orientation estimator, depth z from the DPS, vehicle speeds η_1 from the DVL and vehicle position η_1 from the LBL.

Due to an error in the LBL module no valid data was recorded during the field tests described in section 5. This was only discovered in post-processing after the trials. This means, that for the datasets used in this work only un-stabilized results could be computed. This results in worse results for the GSRL, but since only relatively

4. ALGORITHM

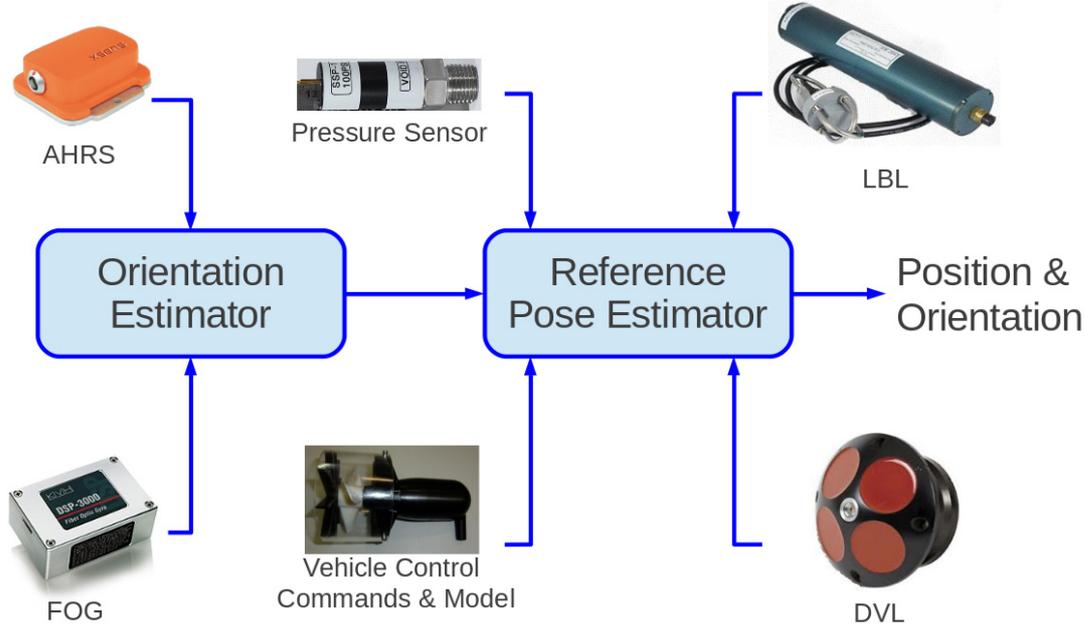


Figure 4.1: Flow-chart of the GSRL implementation on the AUV DAGON.

short trials were conducted (less than 1h), this was deemed acceptable. After the LBL-stabilized trials can be repeated in a future field campaign, this work will be respectively updated.

4.2 SURE-SLAM Algorithm

The AUV DAGON was built with a new localization algorithm in mind. This algorithm is supposed to replace the DVL and LBL sensors of the GSRL in the long run with computer-vision-based measurements using a stereo camera system observing the ground beneath the vehicle (ground relative navigation). This is accomplished by utilization of a SLAM (simultaneous localization and mapping) approach. A flow-chart for this implementation (called SURE-SLAM, Stereo Underwater Realtime Exploration-SLAM) is shown in figure 4.2. Note that besides the LBL and DVL all other components are used from the GSRL in this case, which is not a necessity for the application of the presented validation scheme. Further the visual system can be operated completely alone, with the disadvantage of reduced stability (an experiment for this case is shown in section 5.2.15). With the nomenclature coined in chapter 2 this will be the

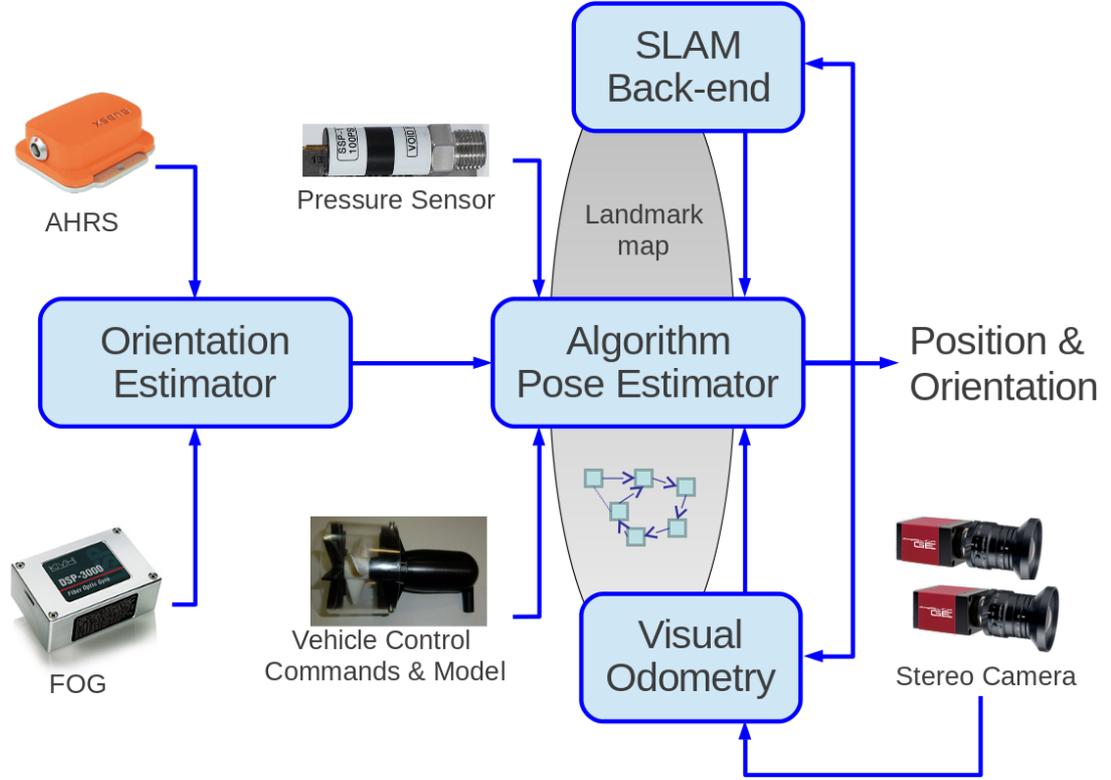


Figure 4.2: Flow-chart of the NLA implementation on the AUV DAGON.

NLA for the rest of this work so the two terms NLA and SURE-SLAM can be used as synonyms for the most part. The localization algorithm has three major parts: visual odometry, SLAM and the Kalman-filter (pose estimator). The visual odometry computes the motion between two camera image pairs (from the stereo camera system) by extracting and matching of SURF-features (5). The resulting information is of similar quality as the DVL measurements in the GSRL, since only changes in motion are computed. Its basic principle is shown in figure 4.3, it is described in detail in section 4.2.1. The SLAM component keeps track of a global feature map and the vehicle's poses in form of a graph. It recognizes when the vehicle passes over a patch of ground it already passed before (loop-closing). As soon as such a loop-close occurs it is integrated into the graph as additional link, reducing the overall uncertainty (graph-based SLAM (21)). This way it can greatly reduce drift-induced deviation and increase long-term stability, similar to the effect of an LBL. The basic idea of this approach is shown in figure 4.4, it is described in detail in section 4.2.2. The pose estimator for SURE-SLAM is using

4. ALGORITHM

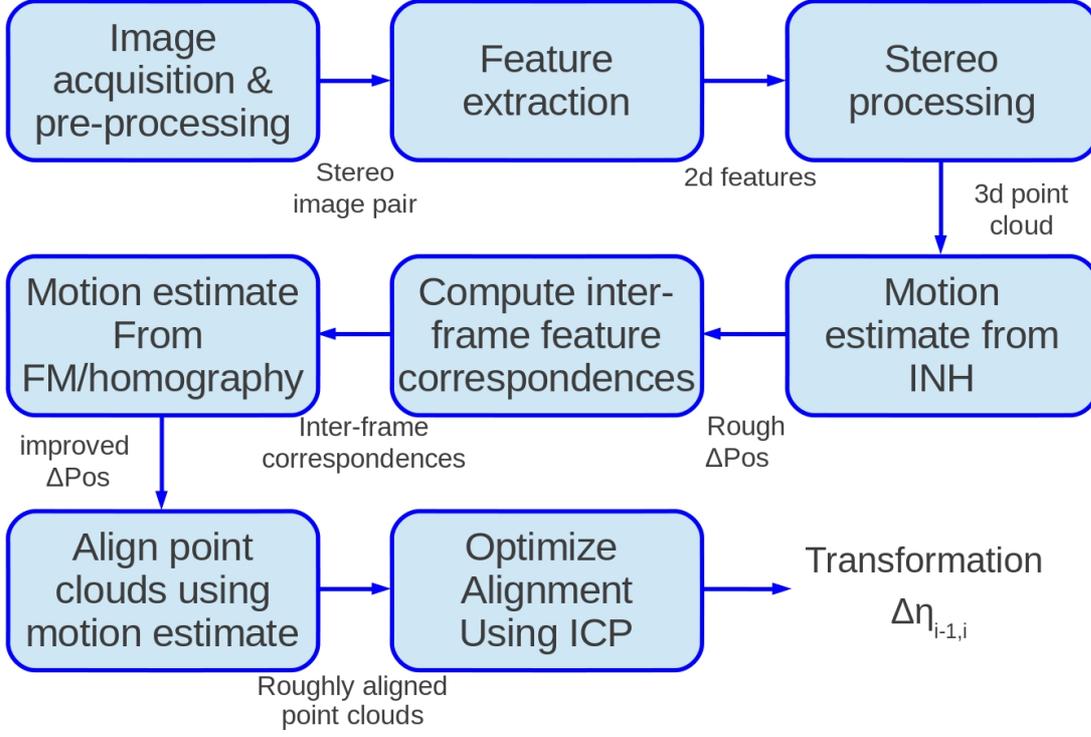


Figure 4.3: The main steps in the visual odometry algorithm.

the same implementation as its GSRL counterpart, only with different parameters and input ports.

4.2.1 Visual Odometry

This section describes the visual odometry component in detail.

4.2.1.1 Pre-Processing

Before the camera images can be used, a number of pre-processing operations have to be executed. The first of these steps is to assure the two images forming the stereo pair have been recorded simultaneously. While on DAGON the two cameras are hardware-triggered simultaneously (see section 3.3.7), it needs to be checked that the two frames currently transported by ROCK belong to each other. This is simply done by timestamp comparison. The next steps are image resizing and de-Bayering. Since the algorithm does not necessarily operate on the full camera resolution, the images have

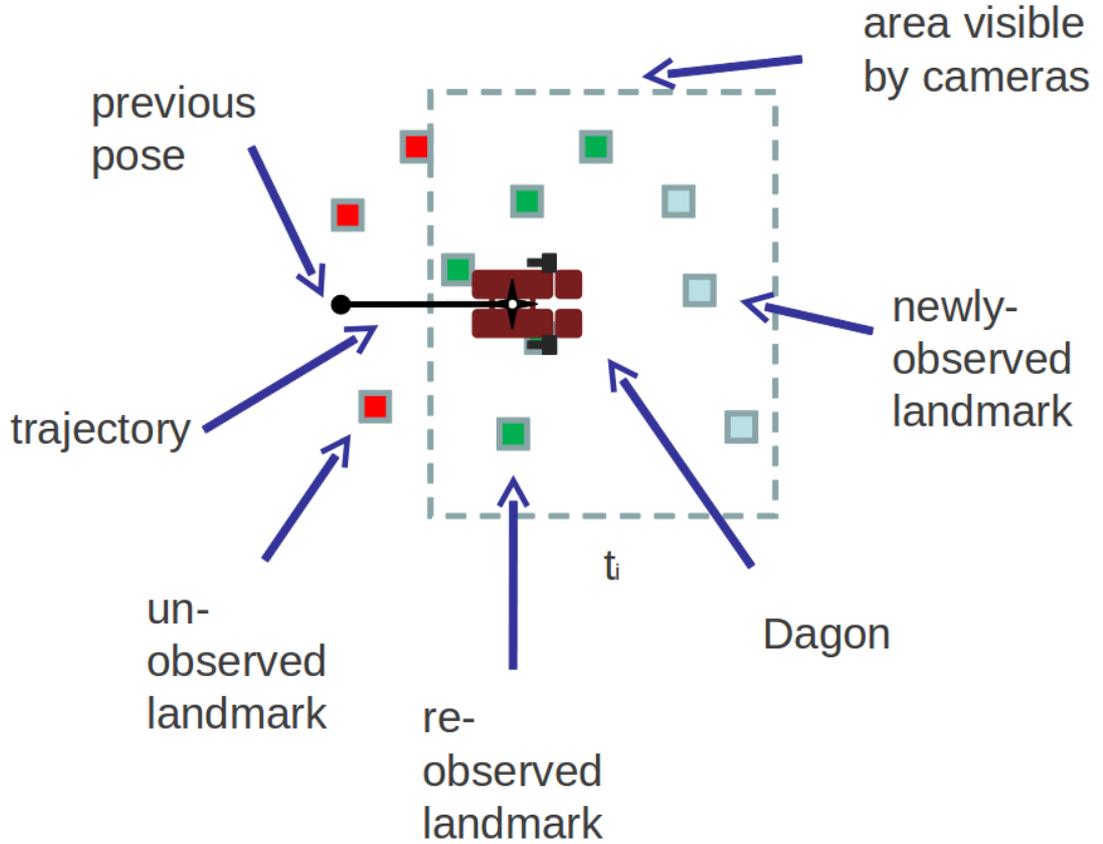


Figure 4.4: The basic idea of a graph-based SLAM approach.

to be resized to the desired resolution (while 1/2 or 1/4 of the original Full-HD resolution can be selected, all experiments described in this work have been conducted with 1/2 HD-Resolution, 960x540 pixels). Additionally the images have to be de-Bayered, since the color information from the cameras is encoded in a Bayer-pattern. The last pre-processing step is image rectification using the known calibration parameters. Rectification (37) assures that all epipolar lines are horizontal and parallel, making stereo matching faster. It also compensates for image distortions.

4.2.1.2 Feature Extraction

A key step in the visual odometry approach is the extraction of spatially and temporally stable salient features from the stereo images in each frame. For this purpose Bay's SURF (speeded-up robust features) feature detector/descriptor is used (5). Based on

4. ALGORITHM

Lowe’s SIFT (scale invariant feature transform) (38) SURF features combine the robustness of SIFT features with strongly improved runtime characteristics. The OpenCV implementation used in this algorithm yields the spatial key-points together with a 128 dimensional descriptor for each key-point. The parameters for the SURF feature extraction are adjusted according to the number of found features in the last image using a simple proportional controller. This assures a relatively constant number of features even with changing scene structure or light and works very well. The number of desired features was selected to be in the order of 400 features for this work. This step results in two lists of SURF features, pl_i and pr_i for the left and right images respectively at frame number i . The actual implementation of the visual odometry allows a number of different detectors/descriptors to be selected (see section 4.2.4.2). For the purpose of this work the SURF descriptor was determined to be the best compromise between speed and resulting quality. Section 4.2.4.2 describes the performance enhancement by computation of the feature extraction on a GPU.

4.2.1.3 Projection into 3D-Space

Using the information of the stereo camera calibration the extracted features are matched between the two stereo images. Non-matched features are dropped, typically resulting in about 250 matched stereo features per image pair in the experiments. Including the information of the camera calibration in this step increases robustness, since it adds an additional constraint to the purely feature-based matching: only two features on the same epipolar line can be valid features. This criteria can be softened by allowing some neighborhood of the epipolar line, since in a real application minor mis-calibration should be tolerable. For this work a neighborhood of 10 pixels was selected as tolerance in this work. This results in the reduced lists pls_i and prs_i . Using the camera calibration again the lists can be combined into a 3D-point-cloud pc_i by triangulation. Additionally the mean altitude A_{mean} of pc_i is calculated and stored for later usage in loop closing.

4.2.1.4 Image Homography/FM Estimation

The image homography describes the geometric relations between two camera images of a planar scene (26). Its computation requires information about the projection of a number of points in the scene (at least 6, as stated in (25)) onto both images,

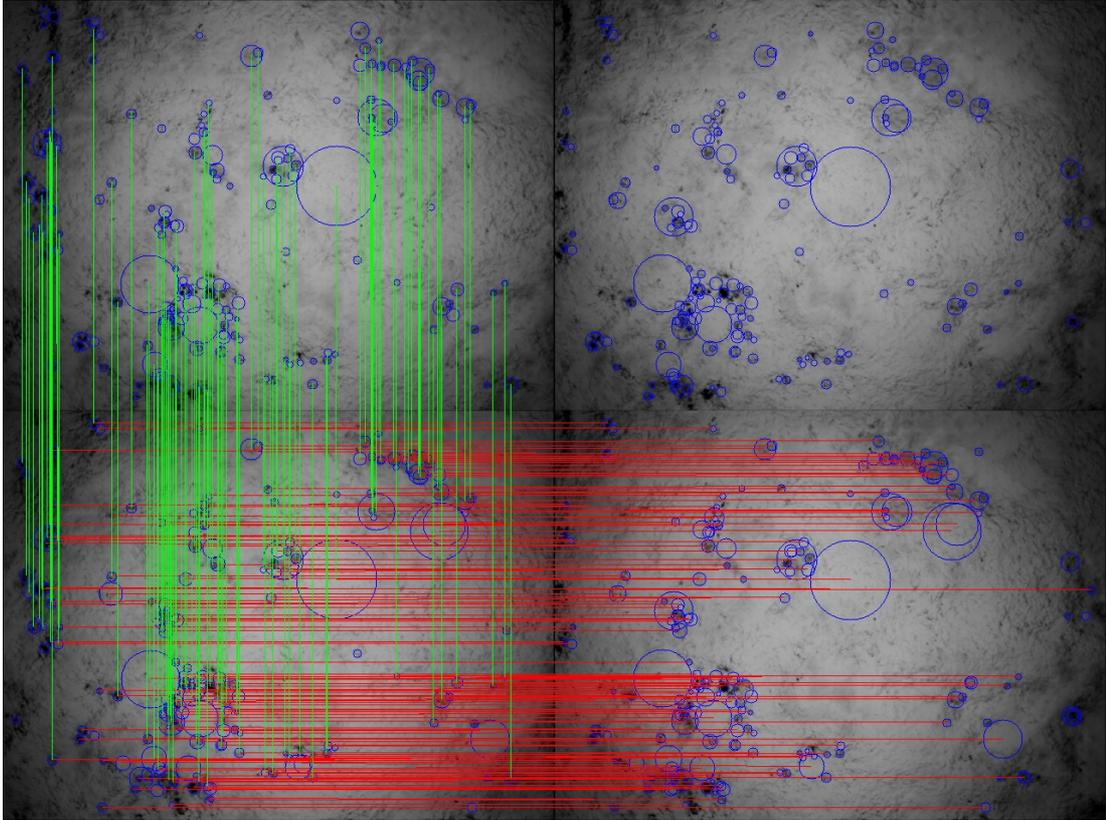


Figure 4.5: The relations inside two subsequent image pairs. The extracted features are shown as blue circles. The stereo correspondences are shown as red lines (only for the lower pair), which are horizontal since the images are rectified. The green lines show the inter-frame correspondences (only shown for the left image pair), which are vertical since the cameras moved in a straight line between the two frames.

meaning that the feature correspondences of these point projections have to be known. This inter-frame feature matching is a key step in the algorithm. Since in real data noise is to be expected (either wrong correspondences or non-accurate points among the images), a probabilistic refinement of the results is advisable and has been used here (RANSAC-algorithm, see (20)). Once the homography is computed the camera matrices can be directly recovered (details see (25)). For the purpose of the algorithm described here the number of points used as input for the computation lies in the order of 100 points. Formally this means that the point lists pls_i and pls_{i-1} are compared to compute the homography h_i describing the transformation from frame $i - 1$ to i . After the computation of the homography all correspondences not obeying its transformation

4. ALGORITHM

are removed from the point lists and treated as outliers. The feature correspondences between two image pairs are shown in figure 4.5. The homography estimation fails if not all points in pls_i and pls_{i-1} are images of a planar environment. In this application the images produced by the cameras of the sea-floor can be compared to aerial photography images. Since the distance between vehicle and ground is large relative to the elevation of objects (e.g. stones, plants) on the ground, the resulting images can be handled as if in a planar environment, so the image homography can be used safely. The problem is that in reality this assumption can be broken (e.g. large objects, cliffs). Since 3d data is available (from stereo triangulation) the fundamental matrix (26) can be used in this case. It is the 3d-equivalent of the image homography and only works in non-planar environments, solving this problem.

4.2.1.5 ICP Optimization

The ICP algorithm introduced by Besl (7) can be used to align two sets of 3D points of the same object. It is widely used to align laser-scans made from different perspectives, e.g. in robot navigation or geometric reconstruction (55). It takes two coarsely aligned 3D point clouds and calculates the 4x4 transformation matrix which minimizes the distance between neighboring points. This is done iteratively, while usually a threshold for near point detection is lowered during the process. This means that during this step the two point clouds pc_i and pc_{i-1} are processed with the ICP algorithm resulting the the transformation t_i^{i-1} which transforms from frame $i-1$ to frame i . Additionally, the ICP computes the back-projection error which is used as covariance matrix for the transformation.

Typical for laser-scans no further descriptive information is available for the individual points in the cloud. This is different in the data available for the algorithm described here: each 3D point is associated with a feature descriptor which has already been matched to the previous frame. Additionally utilizing this information the first iteration of the ICP can use these correspondences as initial guess instead of just relying on Euclidean distances. This has the strong advantage that a good initial match of the two point clouds is established in this phase. The downside is that the feature descriptor is illumination dependent. This means, that in case of re-visiting of a vista the same key-point may have a different descriptor - resulting in a poor matching score. To avoid this the information from the descriptor is only used in the first iteration, all

further iterations use the euclidean distance as sole minimization criteria. Not only can the known feature correspondences be used, but the information from the other vehicle sensors as well. In order to accomplish this the current estimate of the central Kalman filter is used as initial transformation estimate for the ICP. Since the Kalman filter uses the data from the orientation estimator, DPS and control commands, it can give a complete estimate of $\Delta\eta$.

The ICP optimization can be extended to not only consider the last two point clouds pc_i and pc_{i-1} , but the complete history of point clouds. For this purpose, a global point cloud gpc_i is maintained, which is the combination of all prior point clouds without duplicate points $gpc_i \subseteq pc_1 \subseteq pc_2 \subseteq \dots \subseteq pc_i$. The idea of this modification is that the global point cloud gpc_i may contain points which could be matched with the current point cloud pc_i but which is not part of pc_{i-1} . This is typically the case when a loop-close happens. This version of the ICP optimization is called “global” optimization, as opposed to the “local” version described above. During early experiments with real data the “global” ICP optimization was dropped, since it did not scale well with increasing amounts of data.

4.2.2 SLAM-Backend

The SLAM-backend handles the transformations resulting from the visual odometry calculation in form of a graph. Each vehicle pose η_i is interpreted as node in a graph, the edges of the graph being the transformations t_i^{i-1} . The G²O-Framework (General Graph Optimization, (36)) was used for this work as SLAM-back-end. G²O is designed as general framework for optimization of nonlinear least squares problems, SLAM and bundle-adjustment are typical use-cases. Mathematically the SLAM problem is solved by minimizing the function

$$\begin{aligned} \mathbf{F}(\mathbf{p}) &= \sum_{i,j} \mathbf{e}_{ij}^T(\mathbf{p}) \Omega_{ij} \mathbf{e}_{ij}(\mathbf{p}) \\ \mathbf{p}^* &= \operatorname{argmin} \mathbf{F}(\mathbf{p}) \end{aligned} \tag{4.1}$$

The error function

$$\mathbf{e}_{ij}(\mathbf{p}) = \mathbf{e}(\mathbf{p}_i, \mathbf{p}_j, \mathbf{z}_{ij}) \tag{4.2}$$

measures how well the parameters \mathbf{p}_i and \mathbf{p}_j satisfy the constraint \mathbf{z}_{ij} . In the graph-SLAM case \mathbf{p} is a node of the graph (robot pose) and \mathbf{z} is an edge (transformation

4. ALGORITHM

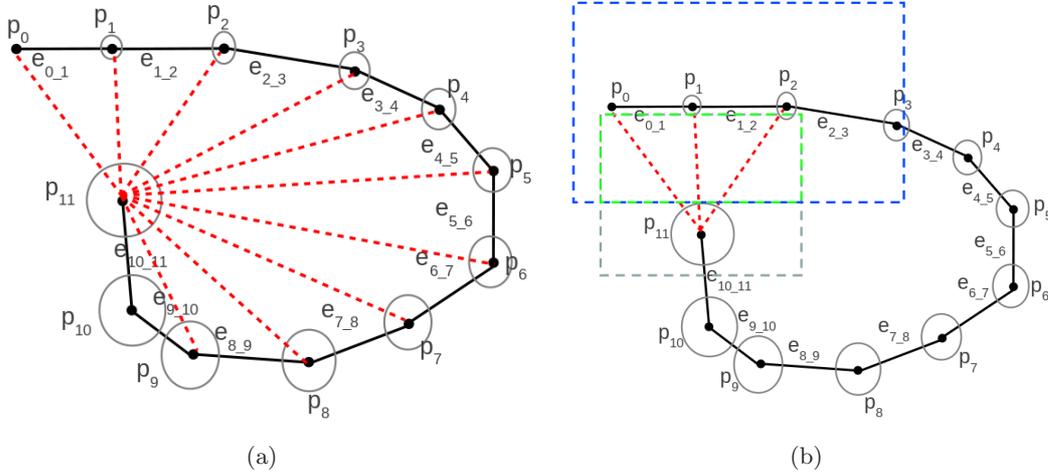


Figure 4.6: Loop Closing. Brute-force approach with n matching candidates (a). Graph-based approach with only 3 matching candidates (b).

between two robot poses). Ω_{ij} is the information matrix (the inverse of the covariance matrix computed in the ICP step).

4.2.3 Loop-Close Detection

The task of loop-close detection is to determine if the vehicle reached previously mapped terrain, and thus can attempt to “close the loop” in the localization graph. This is the reason why the localization structure is not merely a chained list of pair-wise linked vehicle poses (as is the case with pure visual odometry) but may consist of additional dependencies between individual vehicle poses. So formally it is the search for a transformation t_i^m from the current frame i to the loop-closing candidate m .

The first question is how to determine loop closes. Since for the visual odometry a powerful feature descriptor was used (as opposed to faster but simpler point detectors usually utilized), a lot of data is already available for each node in the graph, namely the extracted features. In the visual odometry they are only used between consecutive frames, but since SURF features are size and orientation independent, it is possible to use them to match non-consecutive frames. It could be shown on real data that the matching quality is good enough to use this approach even in sparsely textured terrain (see section 5.2.12). The structure of the visual odometry implementation allows the same code to be used for this type of loop-close checking as for inter-frame

computation: the input point lists are simply changed to be pls_i and pls_m . This approach is computationally very efficient, since it does not require any features to be extracted - all features have been previously computed during the visual odometry processing anyways. If the loop closing is successful, the result is the transformation t_i^m . The basic principle is displayed in figure 4.6.

The next question is how to determine which of the previous frames is a candidate m for loop closing. The simplest and safest way is to always check each new frame against all previous frames. This would result in the list of candidates M to have i members, and thus a very map-size dependent run-time. With calculation times for one loop-close-detection being in the order of 20 ms, even small maps of $i = 100$ nodes would already require seconds for such a computation. The main advantage of this approach is its thoroughness: since M contains all frames, no potential frame can be missed. This approach is called “brute-force” loop closing and shown in figure 4.6(a).

In order to construct M in such a way that its size remains small but at the same time ideally all possible candidates for matching are included, the spatial information of the graph can be taken into account. In order to do this for each node in the graph the imaged area of the sea-floor is computed using the 3D information from the point cloud pc_i :

$$W_i = 2A_{\text{mean}} \tan\left(\frac{1}{2}\text{FOV}\right)$$

FOV is the camera field of view. Now the euclidean distance d_{ij} of two graph nodes i and j is calculated and the virtual visible area of W_i and W_j determined as $W_{\text{max}} = \max(W_i \Omega_{i,\text{max}}, W_j \Omega_{j,\text{max}})$ using the maximum single diagonal element $\Omega_{i,\text{max}} = \max(\Omega_{i,nn})$ of the information matrices of i and j to increase the visible area. The overlap percentage between the two graph nodes is defined as

$$\epsilon = 1 - \frac{d_{ij}}{W_{\text{max}}}$$

Positive values of ϵ are considered loop closing candidates. For loop close candidate detection, initially all nodes are tested for overlap in their portion of surveyed sea-floor. This is a very cheap computation and has been shown to work in previous work (e.g. (18), (39)). When the resulting list M is now sorted for the largest overlap, a test for the first n candidates has a high probability to yield a reasonable loop-close information. A benefit of this approach is that it is only dependent on the length of the loop and not on the total number of nodes in the map. While this approach potentially reduces

4. ALGORITHM

computation times to real-time speed this can still be broken. When the loops get large or the uncertainty is high, i can grow quickly (as was observed in experiments). While the vehicle trajectory can be adapted to keep the loop length low, the uncertainty can only be estimated beforehand. There are a number of ideas on how to handle large candidate lists (see section 6.5.1), but no further implementation on this end was done in this thesis.

Another problem of graph-based loop-closing is that it does not account for so called kidnapping. In kidnapping the robot has to re-localize in a partially known map after being moved to an unknown position in the scene. Mathematically this state can be described by increasing the uncertainty to infinite, effectively returning to brute-force loop closing. A remedy for this problem is the maintenance of a global feature list. This is incrementally built by adding all successfully inter-frame matched features to a global feature list pl_g . This list can then be matched against any frame to return a candidate list of possible frame matches, similar to M . While this process is not as precise as the graph-based loop closing, it solves the kidnapping problem and works significantly faster than brute-force loop closing.

4.2.4 Performance and Optimizations

Since the algorithm is supposed to run in real-time on the AUV DAGON, its computational performance is a key metric for its success. The time required for the individual steps in the algorithm were computed with most of the data described in chapter 5 during different times of the algorithm's development. Most of the components were run-time optimized at some point during the process to improve overall performance. This process will be described in the following sections.

4.2.4.1 Runtime Analysis

The first version of the visual odometry algorithm was not optimized for real-time performance. It was a simple single-threaded C++ application. The main focus in the first implementation was on data quality, not on run-time optimization. This route was taken to avoid optimization of later un-used portions of code. Nevertheless all algorithms were chosen with the future aim of real time-capability in mind. The test data for the first runtime analysis was the synthetic data presented in section 5.1.1. Images were of 640x480 pixel size and loaded from hard-drive instead of captured by

Table 4.1: Average computation times for the major steps of the un-optimized visual odometry algorithm, 640x480 input image.

Pre-processing	Feature extraction	Feature matching	ICP	Loop-closing	Overall
10 ms	224 ms	5 ms	37 ms	20 ms	296 ms

camera. They were also already rectified. Table 4.1 shows the computation times on a 2.67 GHz Intel i7 desktop processor. In order to achieve a performance of at least 10 images per second, the maximum overall duration per calculation is allowed to be 100 ms, so an optimization of at least a factor 3 was necessary. The most work had obviously to be done for the feature extraction step.

The effective speed of the image preprocessing operation (undistort, resize) can be increased significantly by the utilization of a pipelining-architecture, where the images are automatically pre-processed in a separate process (i.e. a different cpu core) while the main process is still finishing computation of the last frame. The run-time of the feature detection and matching is described in detail in the next section 4.2.4.2.

If a successful loop-closure is detected the SLAM-backend needs to optimize the graph. The run time of the SLAM-backend is dependent on the graph size, the number of loop-closures present and the algorithm used for the optimization. For all graphs used in this work $n < 5000$, small number of loop-closures $L < 20$ and both used algorithms (Gauss-Newton, Levenberg) the optimization times were below 100 ms, making it unproblematic for real-time operation.

4.2.4.2 GPU Optimizations

The initial implementation used the OpenCV 2.0 SURF implementation (30) for detection, extraction and matching, which can be improved by utilizing a graphics processing unit (GPU) for this calculation. Cornelis and Van Gool have created a version of the SURF descriptor on a GPU which can exceed frame rates of 10 frames per second (FPS) for images of 640x480 pixels (14). This has the potential of reducing the time required for computation below the 100 ms mark and at the same time offload the CPU. This lead to a GPU based implementation of first the feature detection and

4. ALGORITHM

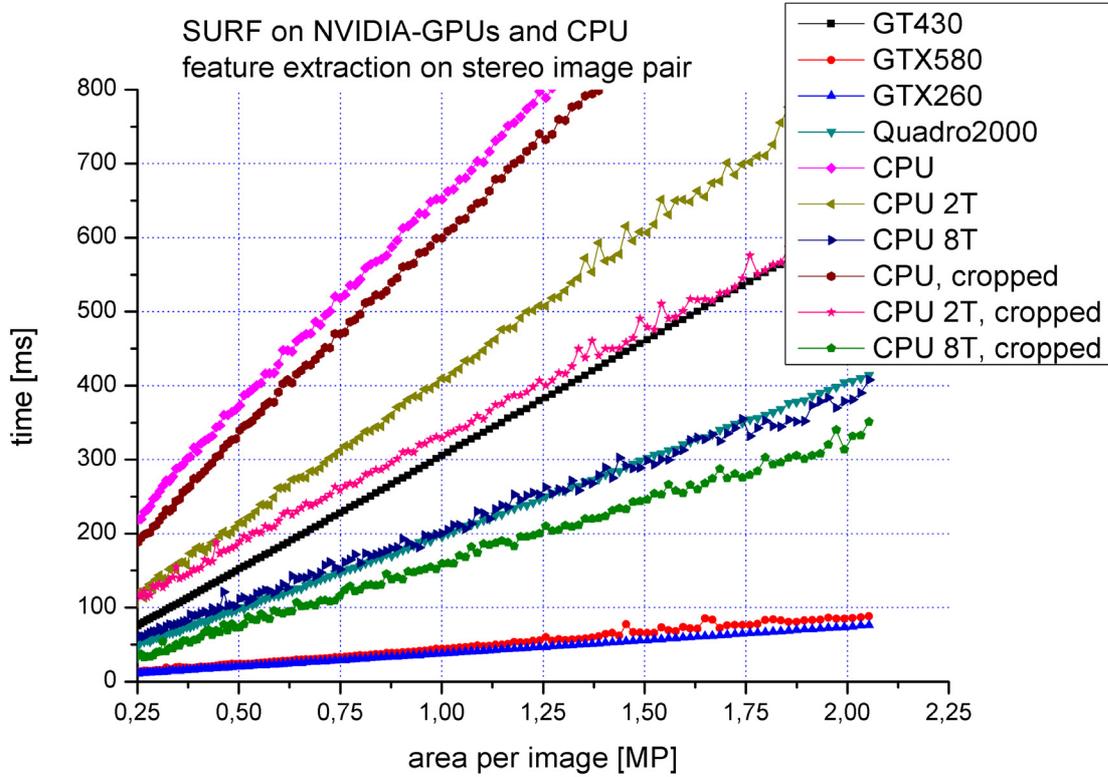


Figure 4.7: Performance of SURF extraction on different generation of CPUs and GPUs applied to differently sized images.

extraction, later the feature matching as well. In order to keep the algorithm versatile, the type of computation is configurable. This allows the algorithm to employ the existing hardware on a given robot. In order to assure this, the complete feature detection/extraction/matching was moved into a separate module which is based on OpenCV 2.3's image feature framework. Besides the method of computation, it allows the selection of the used detection method, implementing not only the SURF detector, but others as well (FAST (52), SIFT (38), MSER (40), STAR (1), Harris (24)). To keep the interface with the rest of the visual-odometry simple, the used feature extractor is always SURF, regardless of the selected detector. The GPU implementation of the feature detector/extractor was tested on a number of different GPUs (see figure 4.7). The smaller of the two GPUs were integrated into DAGON (initially the GT430 was installed, which was later replaced by the Quadro2000), while the GTX260 and GTX580 were used for desktop tests. As expected the performance is only dependent on image

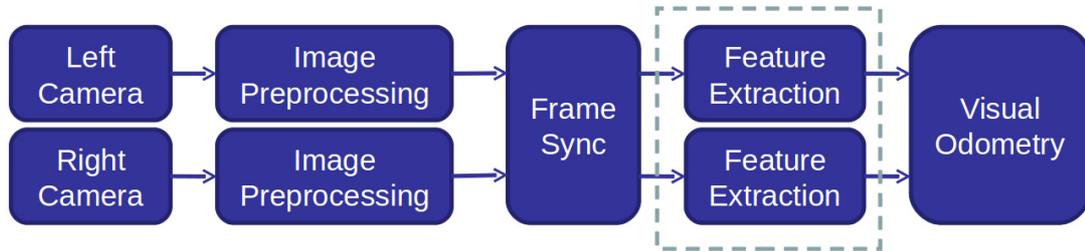


Figure 4.8: Processing pipeline for visual odometry.

area, making this a very sensitive choice. The benefit of using the GPU implementation can be easily see when comparing the extraction time to the CPU time: on a representative sample image of the Rostock dataset (see section 5.1.8 for details on the dataset) with an image area of 0.52 MP (960x540) the CPU required 371 ms, while the GPU only required 150 ms (GT430) or 100 ms (Quadro2000). See table 4.2 for all values on the sample dataset.

4.2.4.3 Multi-Core Optimizations

Some portions of the algorithm can be executed in parallel. Since today’s CPU architectures usually offer multi-core hardware this is a good source for performance improvements. In a stereo camera system it is very intuitive to compute the pre-processing and feature extraction steps in parallel, since up until feature matching they are completely independent. Figure 4.8 shows the simple parallelization pipeline created for this purpose. The reason that the frame-sync step (obviously a non-parallelizable step) happens before the feature extraction but after image preprocessing is simply that it should happen as soon as possible in the pipeline (to avoid computations on non-synchronized images), but could no be placed before the image preprocessing due to data transport reasons of the ROCK framework. The parallelization was done by creating two separate threads for the individual images, which were then executed in parallel, utilizing both CPU cores available on the vehicle DAGON. The parallelization effort can be continued by separating each individual image into sub-images and creating individual threads for these images as well. For this case the sub-images need some image overlap to avoid feature-free zones at the boundaries. While DAGON did not have the CPU cores to test this properly, the respective test was executed on a desktop PC with 8 cores. Each individual image was divided into four sub-images, resulting in

4. ALGORITHM



Figure 4.9: The location of all detected and stereo-matched features on all left images of the Rostock test sequence (approx 4.500 images) plotted as greyscale image. Dashed red line shows the cropping limit (130 pixels).

8 threads in total. The results of the parallelization efforts are shown in figure 4.7 and in table 4.2.

The last optimization implemented resulted from an analysis of experimental data from the Rostock-trials (see section 5.1.8 for details). The location of all detected and stereo-matched features on all left images of one of the test sequences (approx 4.500 images) was plotted as grey-scale image (see figure 4.9). The resulting distribution shows two distinct artifacts: the first is an empty area on the left border, the second is a circular area to the left bottom. The first artifact is a result of the limited stereo overlap, which decreases at low distances to ground. This effect can be used for optimization, since no feature extraction needs to be done in this area. This “cropping” reduces the image by 130 pixel columns, reducing the image area by 14%, and thus directly improving processing time. The second artifact actually was the result of a grease smear on the viewport of the right camera. This grease resulted in a blurring of the affected image area, which reduced the number of detected features.

Table 4.2: Processing speed for feature extraction on a 960x540 stereo image pair on different hardware with different optimizations.

optimization	processing time (ms)	speed-up
CPU, none	371	
GPU, GT430	150	-60 %
GPU, Quadro2000	100	-73 %
GPU, GTX580	29	-92 %
CPU, cropping 130	334	-10 %
CPU, 2 threads	195	-47 %
CPU, 8 threads	84	-77 %
CPU, 2 threads + cropping	178	-52 %
CPU, 8 threads + cropping	79	-79 %

4.2.4.4 Memory requirement

In a graph-based SLAM approach all data is contained in the nodes. In this implementation each node consists of the following data structures:

- Pointer to stereo image pair (recorded onto HDD)
- Two lists of extracted 2d features (≈ 135 kB)
- List of stereo correspondence indices (≈ 1 kB)
- 3d point cloud (≈ 4 kB)
- List of inter-frame correspondence indices (≈ 1 kB)
- Pointer to previous node
- Transformation from previous node
- Covariance matrix for transformation

The resulting overall node size is ≈ 150 kB. With DAGON's current 8GB of memory on its visual processing PC, this results in up to 3 hours of operation at 5 Hz processing

4. ALGORITHM

speed. After this time has elapsed some mechanism for partial storage of the node information has to be devised.

5

Experiments

One of the major tasks during this thesis was the design, preparation and execution of experiments designed to test the feasibility of the algorithm in diverse environments and conditions. Over three years, more than 500 hours were spent in the water with the vehicle, roughly half of it conducting experiments. A total of 3 TB of data was recorded and evaluated.

5.1 Testing Environments

In this section the different testing environments and their specifics are described. The individual experiments in these environments are described in section 5.2.

5.1.1 Synthetic Data

The synthetic datasets were created using a state of the art modeling, animation and rendering program (3dsMAX) providing realistic data and ground-truth position information. The scene used as basis for all input data is shown in figure 5.2. In a first step

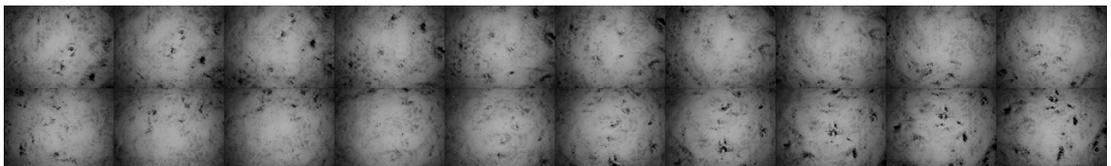


Figure 5.1: A sequence of images from the test dataset showing one of the u-turns. It shows 20 frames from the left camera, using every 10th frame.

5. EXPERIMENTS

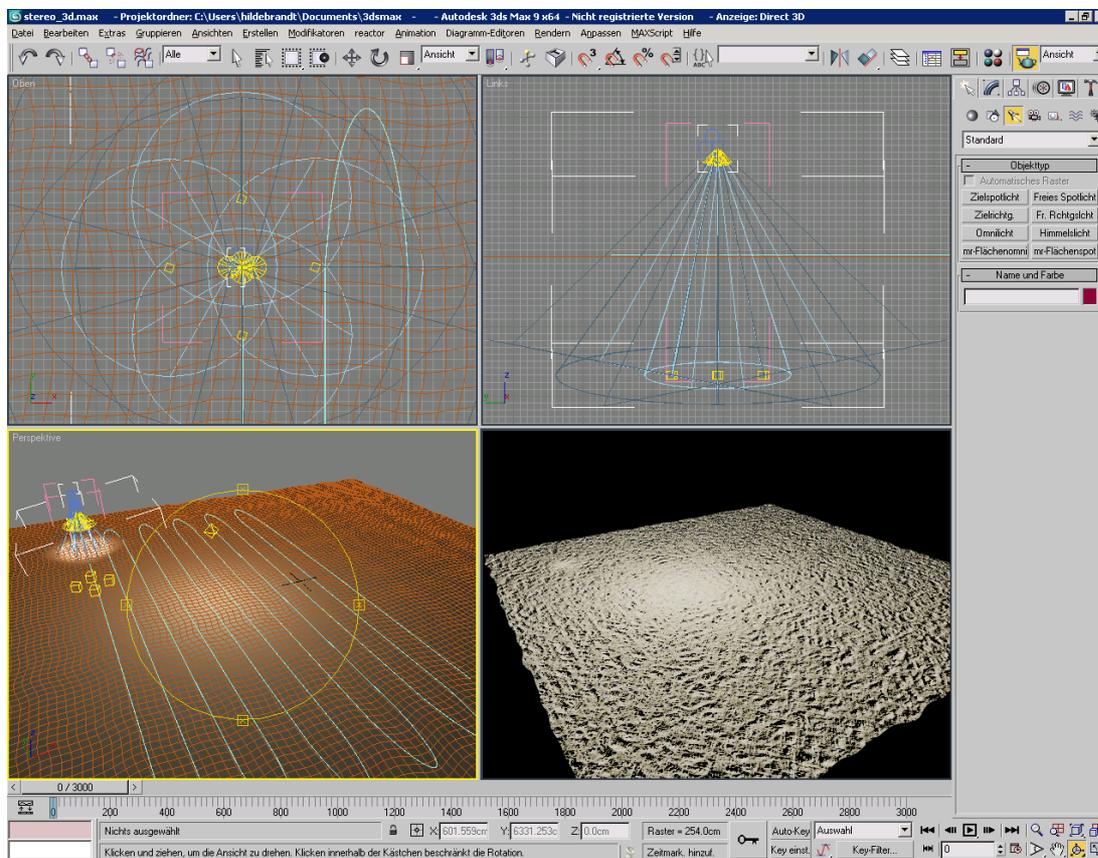


Figure 5.2: The scene used to create the synthetic dataset. Note that an additional omni directional light was added to create this view of the scene.

the impact of different lighting conditions was evaluated and a configuration with four discrete lights illuminating the scene homogeneously (especially at the border areas) was chosen. Two datasets were produced: one dataset with no visual noise (from now on referred to as “plain” dataset) in the images and a second dataset with artificial marine snow (referred to as “snow” dataset). The marine snow is simulated by five layers of randomly moving particles of varying size and speed, each layer consisting of 100.000 particles of which approximately 1200 are visible in each image. This closely resembles real marine snow effects under calm sea conditions. The rendering of the datasets took about a month on a 2.67 GHz Intel i7 PC. Each dataset consists of 2x3000 images in the resolution 640x480 pixels taken from two virtual cameras with a field of view of 60° (a short sequence of these images is shown in figure 5.1). The baseline of this virtual stereo camera set is 32 cm, yielding an approximately 90 % image overlap at

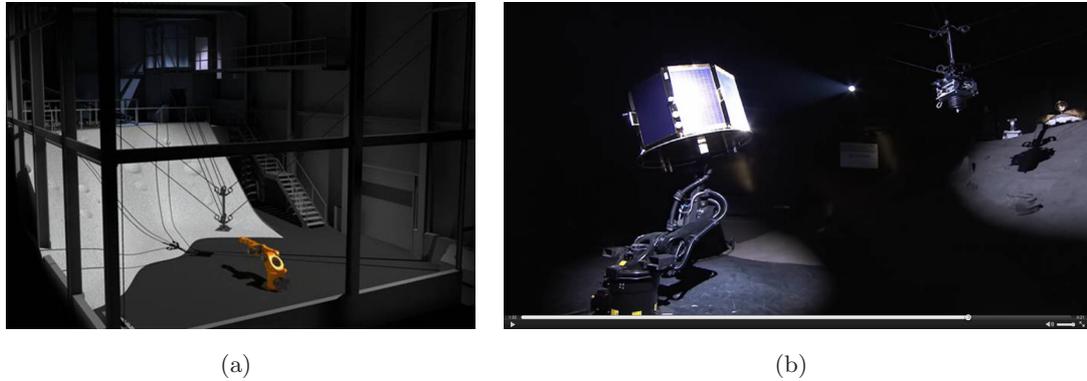


Figure 5.3: Computer rendering of the space exploration hall at the DFKI RIC (a). View of the crater with Spider-Cam and Kuka-satellite-simulator (b).

working distances. The virtual cameras and lights were traveling in a survey pattern, meandering in five consecutive s-shapes over an area of 400x250 m, covering a distance of approximately 4.5 km. The mean distance from the floor was 4 m. The resulting inter-frame overlap was about 96 %. On a real vehicle this would reflect image processing rates in comparison to vehicle speed, and thus represents an important practical factor.

5.1.2 Space Exploration Hall

The space exploration hall is a special laboratory at the DFKI-RIC which was designed as testing environment for space exploration robots. With a length of 24 m, a width of 12 m, and a height of 10 m is is a large test area for robotic vehicles. All surfaces of its interior are kept black in order to simulate the emptiness of space. It is equipped with a number of environments, testing equipment and measurements systems. Its most prominent feature is the large crater structure occupying most of the central area. With an area of 105 m² and an inclination of between 25° and 45° it offers a tough testing environment for legged-robots. The two systems permanently installed in the hall are the Spider-Cam, a cable-guided motion platform, and the Kuka-satellite-simulator (shown in figures 5.3(a) and 5.3(b)). The hall is equipped with a sophisticated lighting system, able to simulate varying degrees of strong, directional sunlight (as is to be expected on thin and no atmosphere planets). A Vicon 3D-Tracking system enables referenced robot experiments.

5. EXPERIMENTS



Figure 5.4: A view of the DFKI-RIC's underwater testbed with the glass basin in the foreground and the black basin at the back.

5.1.3 Black Basin

The black tank is one of the two test basins in the underwater testbed. It is the smaller of the two ($3.4 \times 2.8 \times 2.5$ m, 20 m^3) and has only two small windows. Its purpose is to allow visibility control by preventing any external light from entering and the ability to select the turbidity of the water. This was very useful to determine if the illumination of the vehicle was potent enough and to find the limit of turbidity the visual mapping algorithm can still cope with.

5.1.4 Glass Tank

The second test basin in the DFKI-RIC's underwater testbed is the 40 m^3 glass wall tank, which is a $5 \times 4 \times 2.5$ m glass/steel basin. Due to its three large glass sides it is ideal for basic testing and low-level controller tuning since the vehicle can be observed very well. Its moderate size allowed even some scientific experiments under very controlled conditions. The ground is covered with 16 mm to 32 mm sized gravel, which presents

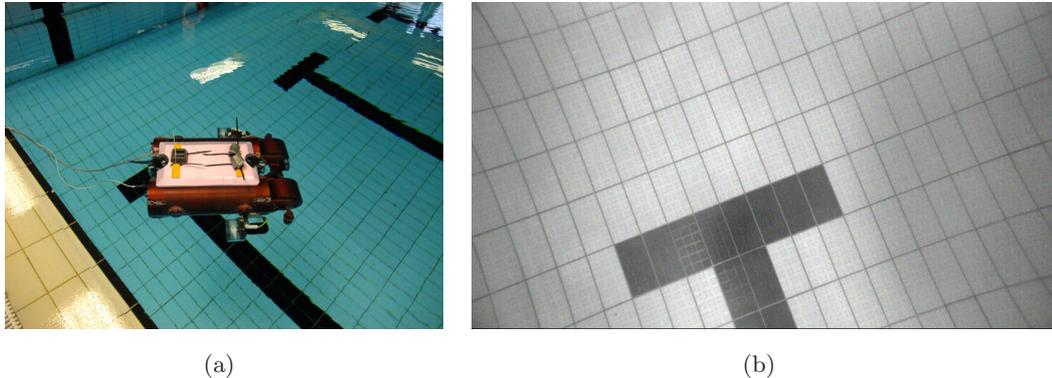


Figure 5.5: DAGON in the University Pool 2010 (a). Camera view of the floor of the university pool. The camera’s distortion is well visible in the tile pattern (b).

a very good substrate for both the reference localization (especially the DVL) and the cameras.

5.1.5 University Pool

The university pool is a swimming pool at the University of Bremen. It was kindly allowed by the University to conduct experiments in this pool in a fixed time-frame. The pool has a size of 30x17x2.5 m, tiled walls/floor, fresh (chlorinated) water and an underwater illumination system. It was ideally suited for controlled larger-area experiments with the vehicle, as well as experiments in repetitive environments. Since the available time-frames were only short, more complex experiments could not be conducted in this environment. Figure 5.5 shows DAGON in the pool during an early experiment as well as a sample camera frame of the tiled ground.

5.1.6 Unisee

Near the University of Bremen a quarry pond called “Unisee” (German for university lake) is conveniently located only 5 minutes of driving distance from the DFKI-RIC. This lake was an important testing environment for both the vehicle and the algorithms developed on it. Easy to access all-year and with a number of natural features it was a major testing ground for preparation for the experiments in the more complex and demanding environments. A life-saving station at the southern edge allowed the DFKI-RIC’s boat to be watered and thus supervision of the experiments from above water.

5. EXPERIMENTS

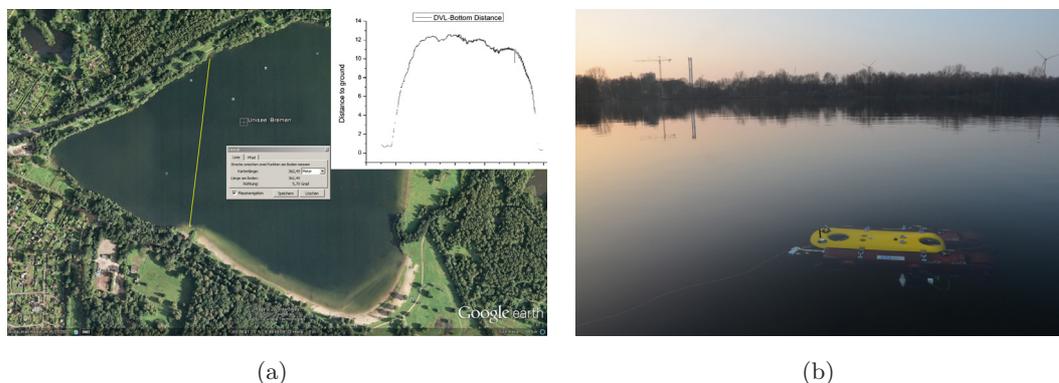


Figure 5.6: The lake near the university, called “Unisee”. Satellite view of the lake with depth profile (a). DAGON surfacing at the end of an experiment in the evening (b).

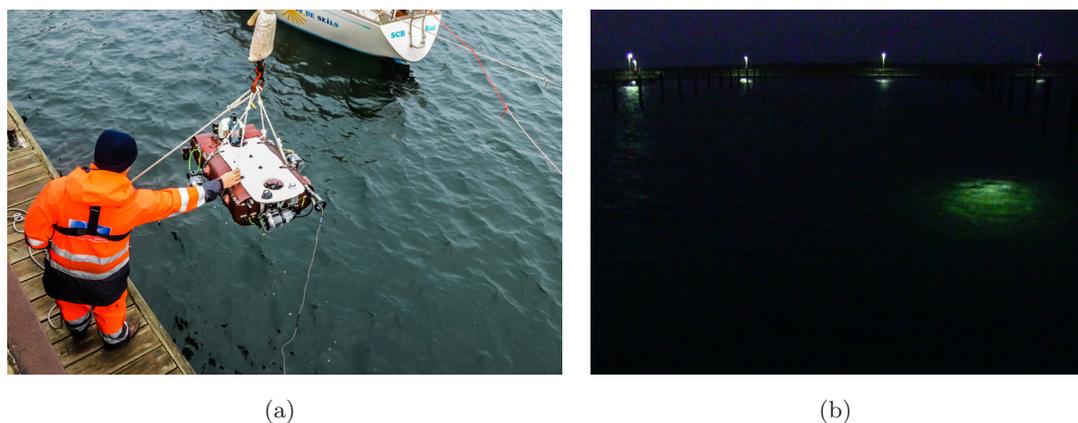


Figure 5.7: Deployment of DAGON in the harbor in Kiel, Winter 2011/2012 (a). Night-time experiment with DAGON visible as bright area at the right border of the photo (b).

The lake is a former gravel pit, which makes its shores very steep. Unfortunately, this also means that its ground is muddy, reducing visibility to one meter and below most of the year. Nevertheless the lake was a versatile test environment for gaining operational handling experience in an outdoor environment, as well as for large-scale experiments such as the trajectory follower. A satellite image of the lake together with a depth profile is shown in figure 5.6.

5.1.7 Kiel

As preparation for the tests in the Baltic Sea, which were planned for mid-2012, a series of tests were conducted in November 2011 in a harbor near Kiel. Originally it

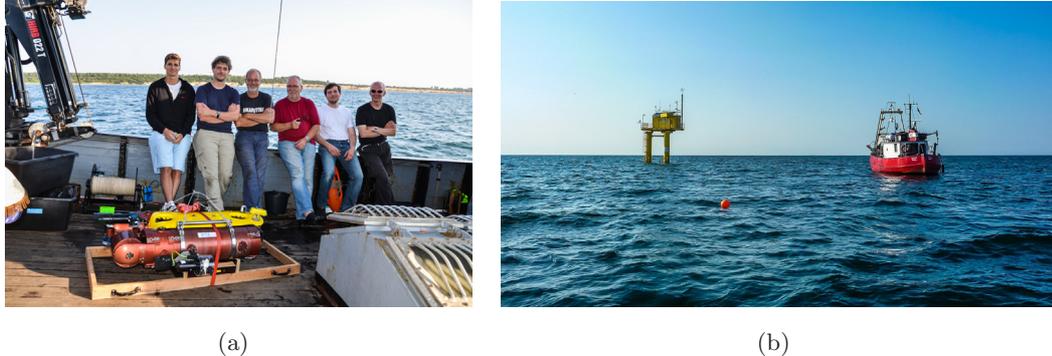


Figure 5.8: Team photo of the sea-trials summer 2012 in Rostock, taken on the R/V Gadus on the last day of the trials (a). The testing environment at the artificial reef Nienhagen ($54^{\circ}10.5'N$ $11^{\circ}56,6'E$). To the left the observation tower can be seen, to the right the research vessel *Gadus*, which was used as base (b).

was planned to only use the harbor as access to the Baltic Sea, but due to bad weather and strong wind the experiments were forced to stay inside the harbor. Still a lot of preparation could be done, as the visibility was better than in the lake, and DAGON and its operators had to cope with waves, obstacles and bad weather.

5.1.8 Rostock

The final tests were conducted mid-2012 in the Baltic Sea. The artificial reef off the coast of Nienhagen¹ near Rostock was the test environment. The artificial reef is a test site for biological long-term observation, where 1400 artificial concrete elements were placed in a $50\,000\text{ m}^2$ area. For this campaign the RV *Gadus* was chartered from the University of Rostock as base vessel, and additionally brought the DFKI-RIC's own small RV, the 6 m RIB "Polarsternchen" (see figure 5.8(b)).

This test site was the first time DAGON was used in the open sea. In mixed weather conditions (from sunny to strong rain, calm to 5 Bft), current and waves, the vehicle performed very well. Water conditions and visibility were optimal, a huge amount of data was recorded. On the last test day of this two-week campaign, DAGON performed two fully-autonomous missions, without any cable attached.

¹<http://www.riff-nienhagen.de>

5. EXPERIMENTS

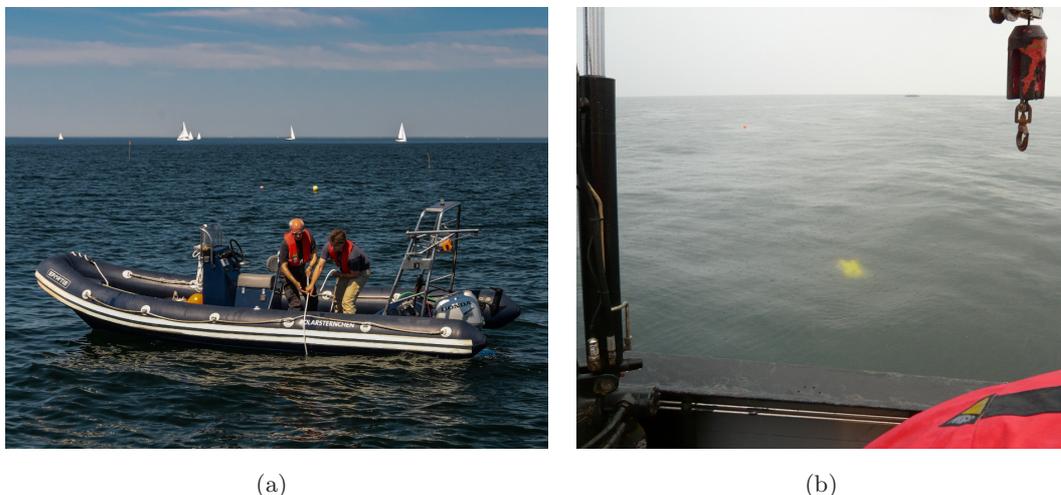


Figure 5.9: Experiments at the artificial reef in Rostock, Germany Summer 2012: Deployment of the LBL system from a small RBI (a). The vehicle in the water (b).

5.1.8.1 LBL-failure

During the trials in Rostock the LBL system was deployed and its data recorded together with the other sensors as part of the GSRL system. During data analysis after the trials it became apparent that due to a defect of the vehicle transponder, most of the data recorded after day one was faulty. The effect was that in the central GSRL Kalman-filter the LBL readings were rejected - since they did not contain any useful data a correct reaction. This was not noted during the trials since the impact of a non-functional LBL on the GSRL was minor for the relatively short individual experiments. Further the focus was put on the SURE-SLAM algorithm – after initial LBL-survey and testing the LBL sensor did not get the attention required. The impact on this work is a reduction of the GSRL quality. Because the LBL is responsible for long-term drift compensation, the maximum error of GSRL measurements used here is unbound, being only dependent on relative measurements by gyroscopes and DVL.

5.2 Description of Individual Experiments

In this section the individual experiments will be described. Each section is named and has the environment in which the experiment took place in parentheses behind the name.

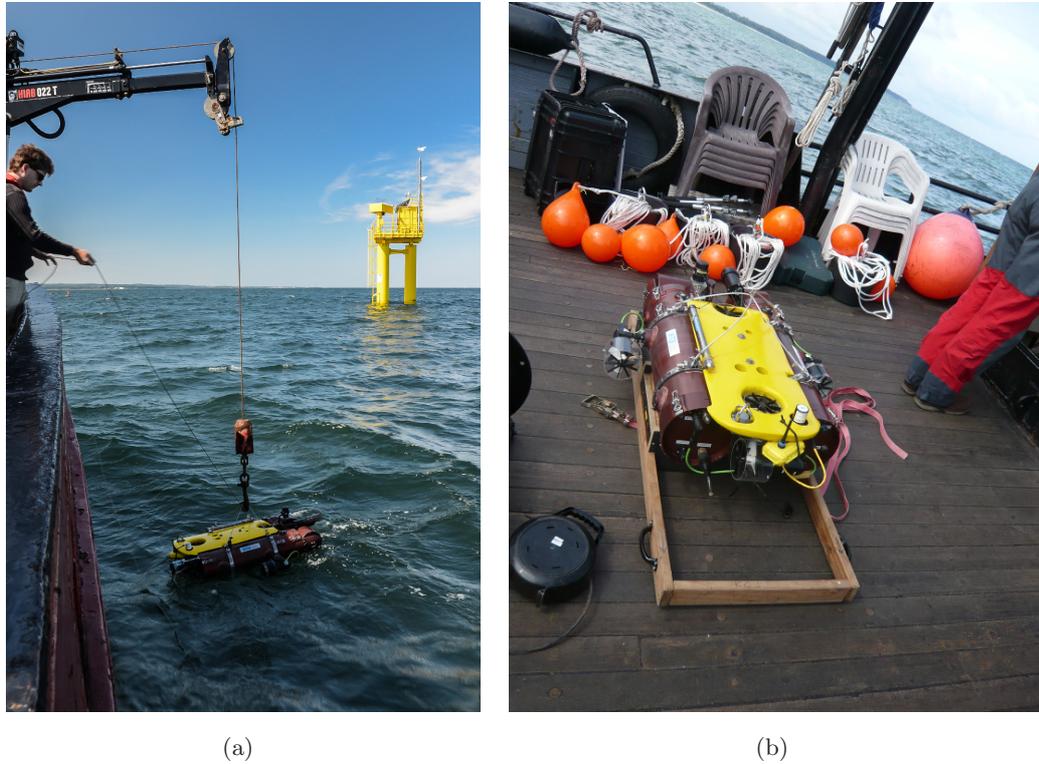


Figure 5.10: Experiments at the artificial reef in Rostock, Germany Summer 2012: deployment of the vehicle from the support ship (a). Preparation of the vehicle and LBL system on deck in the left figure (b).

5.2.1 Meander without Crossings (Synthetic)

The overall results on the synthetic datasets were very promising. The absolute error after traveling 4.5 km lies in the order of 5 m (see figure 5.11(b) for a graphical representation). Most of this error originates from poor orientation estimation, especially at the turning points of the trajectory. One reason for this behavior can be found in the generation of the synthetic dataset: While the translational speed of the vehicle is kept constant, the rotational speed is not limited. This results in very high rates of turn at the turning points, and as final consequence in poor estimation. Future datasets will take care of this effect by limiting rate of turn. This fact will also impact the control of the real AUV. Unfortunately, the fact that all rotation is captured only by 20 images (at each turning point) made a variation in image density impossible - the intended usage of only every 10th image would have resulted in only 3 images used for a 180°

5. EXPERIMENTS

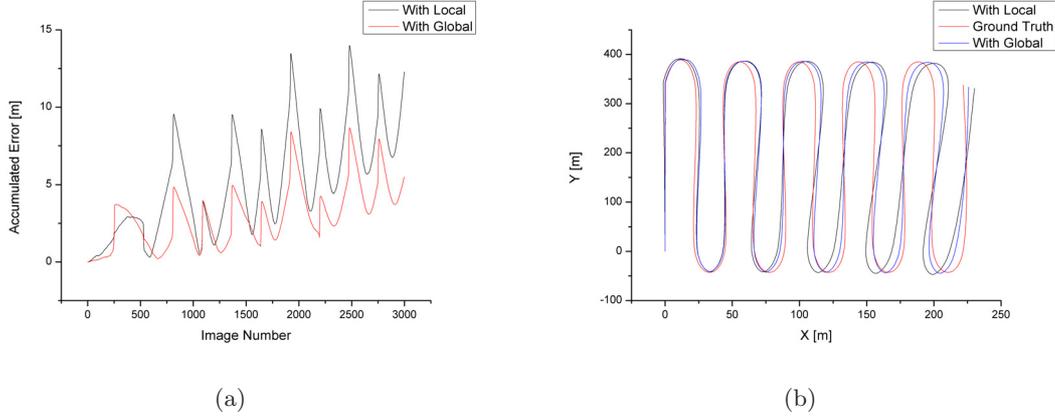


Figure 5.11: Distance error between visual odometry and ground truth for local and global algorithm variants (a). Comparison between the performance of visual odometry with and without a global map (b).

rotation - which is not feasible.

Typical for a visual odometry approach is the unbound cumulative error (see figure 5.11(a)). The sparse relative deviation of $d_{s,r} = 0.1\%$ still is very low, about 50 times lower as in (13), (41) and (47). Since they were operating in a real environment, higher error rates are to be expected, but an improvement of this magnitude is still a good starting point.

One of the key properties of the algorithm became apparent when comparing the results of the 'plain' dataset with the results from the 'snow' dataset. The noise has virtually no effect, the accuracy stays coherent in all measured quantities. This behavior is explained first by the feature detector/descriptor, which rejects small features. The second explanation is the ICP's ability to iteratively reject points which do not fit the overall transformation estimate by decreasing the threshold value for the local search.

The deviation of the start to end position was interpreted as deviation over the driven distance. The results were a deviation of $d_s = 4.51$ m after a traveled distance of $|\eta| = 4500.0$ m, resulting in a relative deviation of $d_{r,s} = 0.1\%$. Since a ground-truth measurement was available, a standard deviation could be computed for the error between the ground-truth measurement and the visual odometry measurement. This standard deviation was as low as $\sigma = 3.317$ m. All measurements were taken for the "Plain" dataset.

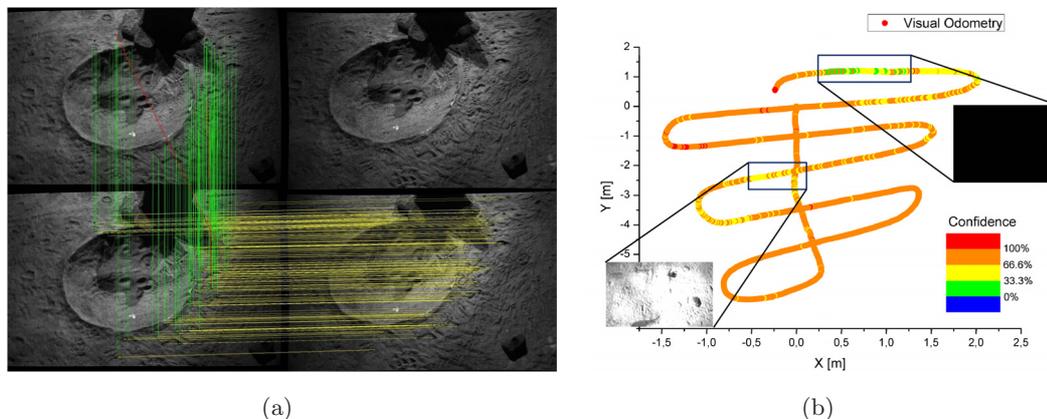


Figure 5.12: Geometric relations of two consecutive stereo pairs with features recorded during the spider-cam trials (a). Reconstructed trajectory of the spider-cam trial with evaluation of low-quality regions (b).

5.2.2 Spider-Cam (Space Exploration Hall)

One of the early real-world datasets came from a recording of stereo data from the spider-cam system in the space exploration lab at the DFKI-RIC. The spider-cam system is a cable-guided platform which can traverse the complete space exploration lab in all three dimensions while carrying up to 150 kg of payload. For these trials it was equipped with a stereo camera system facing downwards and guided with a survey trajectory over the artificial moon crater in the exploration lab. The benefit of this trial was to receive real-world data before the completion of the AUV DAGON, in a manner which allowed high-quality reference localization information to be recorded simultaneously (the spider-cam system has a localization accuracy of 1 mm) and with sparsely structured image data. The results were very promising and showed that the approach was feasible for real-world data (see figures 5.12(a) and 5.12(b)). The resulting deviation between the spider-cam measurement and the visual odometry was $d = 0.6$ m, after a traveled distance of $|\eta| = 50.0$ m. A relative deviation of $d_r = 1.2\%$ was calculated. The standard deviation of the error between the two estimated trajectories was $\sigma = 0.32$ m. This first result with real image data, although not yet in underwater conditions, showed the feasibility of the approach. At the time of the experiment the SLAM component was not yet available.

One very important problem of the setup became apparent with this experiment:

5. EXPERIMENTS

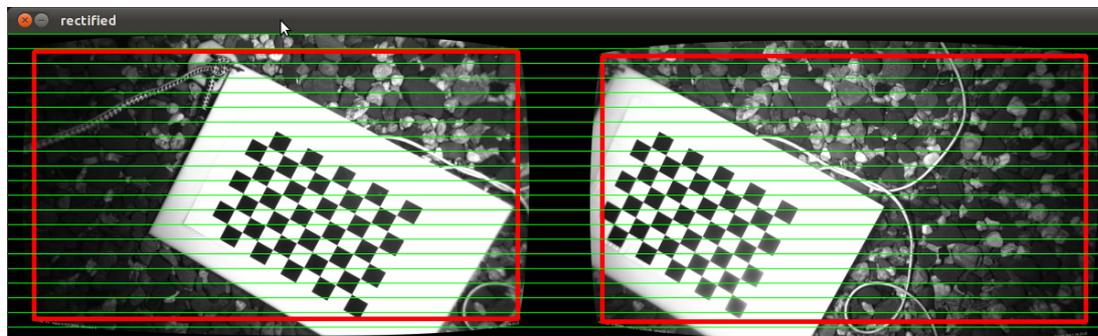


Figure 5.13: Results of camera calibration in the underwater testbed.

Since the crater was very heterogeneously illuminated, the camera (operating with a fixed exposure value) had regions of under- and over-exposure. The necessity to run the stereo camera system in auto-exposure mode became clear, which required the scheme described in section 3.3.7 to be implemented.

5.2.3 Camera Calibration (Glass Tank)

One of the first tasks which needed to be done when moving to real cameras was camera calibration. For the underwater datasets the stereo camera of DAGON had to be calibrated. The quality of calibration is of key importance for good results with the visual system, so it had to be done with care. For the calibration, a chessboard calibration pattern was glued onto a polyethylene sheet, which was anchored to the ground of the glass tank. Then DAGON was moved over the calibration pattern until 11 views of the pattern were recorded by both cameras: centered at maximum distance, centered at working distance, centered at minimum distance, pattern in all four corners at working distance, centered and tilted by 30° in both main image axes at working distance. This set of 22 images was the input dataset for a calibration according to Zhang (67), using the OpenCV implementation for stereo calibration. This resulted in both intrinsic as well as extrinsic calibration parameters for DAGON's cameras. Due to a construction error (see section 3.3.7 for details) the calibration had to be repeated before each major experiment. Theoretically it also had to be repeated when migrating from fresh to sea water. The changes in salinity affect the refractive index of water and thus affect the projection of the objects onto the camera chip. The resulting changes have been experimentally determined to be negligible, allowing the calibration to be

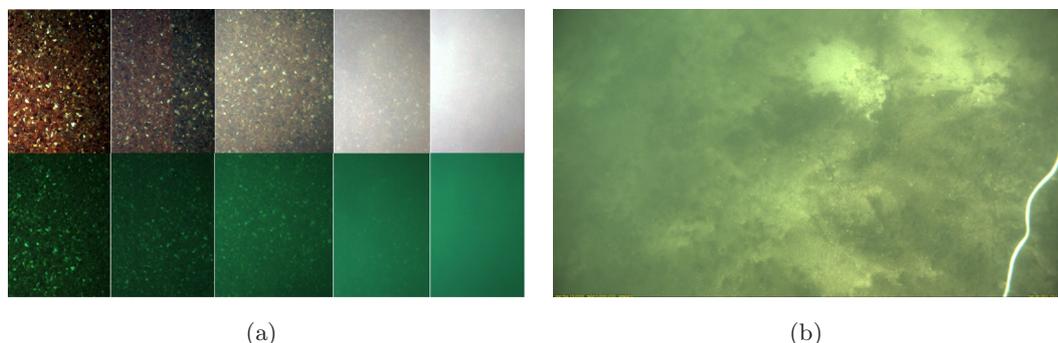


Figure 5.14: Measurements with increasing image turbidity (a). Effect of turbidity in a real environment. The floor in this image is slanted, the approximate image distance in the lower-right corner is 1.5 m while in the upper-left corner it is 3 m (b).

conducted in the safe environment of the glass tank instead of the open-water of further tests. A sample frame from the calibration procedure showing the chessboard in both images after rectification can be found in figure 5.13.

5.2.4 Turbidity (Black Basin)

Since turbidity is a problem for all vision-based sensor systems, its effect on the algorithm needed to be evaluated. Ideally, a complete set of trials would be conducted in different turbidity conditions with the vehicle conducting controlled trajectory following and evaluation of accuracy as described in section 2.2.4.4. Unfortunately, no test area where this could have been realized was available for testing. While experiments were conducted during different degrees of turbidity (see for example figure 5.14), these tests can only be seen as sparse evaluation of this property. To remedy this, a static turbidity series was conducted and only the performance of the feature extractor evaluated. Since feature extraction is the key step in environment interaction, this should give an estimate of the impact of turbidity.

At the small test basin in the DFKI-RIC, turbidity measurements can be conducted. A Seapoint brand turbidity meter measures turbidity in the range of 0-10 FTU (formazin turbidity units (61)). Using the filter pump in closed-circuit mode, clay dust is added into the basin and distributed by the pump. The amount of clay determines the level of turbidity. For this experiment, the AUV DAGON was fixed into the basin with a number of ropes. The tank’s bottom consists of gravel. Three parameters were

5. EXPERIMENTS

Table 5.1: Effect of turbidity to feature stability.

Turbidity	Distance	Light	Stable Features	Stable Stereo Features
0	2 m	off	227	183
0	2 m	on	231	181
0	1 m	off	215	168
0	1 m	on	241	204
2.6	2 m	off	232	192
2.6	2 m	on	194	177
2.6	1 m	off	228	165
2.6	1 m	on	239	186
5.3	2 m	off	160	112
5.3	2 m	on	138	99
5.3	1 m	off	205	160
5.3	1 m	on	187	156
7.9	2 m	off	48	39
7.9	2 m	on	0	0
7.9	1 m	off	169	135
7.9	1 m	on	144	110

modulated: distance to ground (2 m, 1 m), vehicle illumination (on/off) and turbidity (0-8 FTU, 4 measurements). For all 16 individual measurement points two properties were measured: the number of stable detected features from a single camera as well as the stable detected and matched stereo features. A feature was considered stable when within ten frames there was at most one frame where it was not detected. The camera images with the different levels of turbidity are shown in figure 5.14, the resulting measurements are summarized in table 5.1.

5.2.5 Repetitive Environment (University Pool, Norway)

One of the challenges for the feature-based localization algorithm is repetitive structure since distinctive correspondences between features cannot necessarily be established. The repetitively tiled floor of the university pool was an ideal environment to test the



Figure 5.15: Camera view of the fish net in Norway.

performance of the algorithm under these conditions (see figure 5.5). A second experiment was conducted on data recorded in a fish-farm in Norway, where DAGON was surveying a fish net (see figure 5.15). In both experiments the visual-odometry component was working as long as the internal navigation hardware was used to stabilize the measurements (see section 5.2.15 for details). The number of matched features was lower than in most other experiments, since due to self-similarity a high number of wrong feature matchings were present in the data. These were removed by the homography/fundamental matrix computation. Loop-closing however was impossible under these conditions. Only the inclusion of additional landmarks in the scene (numbers on the pool floor for the pool dataset and anchor-lines for the Norway-dataset) resulted in successful loop-closing.

5.2.6 Sparse Validation of Reference Measurement (Glass Tank)

This section describes the sparse indoor validation of the GSRL. As stated in section 2.2.2 since the GSRL is used as reference localization measurement, it could only be sparsely validated. To accomplish this, two experiments were conducted: First the trajectory-follower module was used to drive a rectangular trajectory in the 5x4 m test-

5. EXPERIMENTS

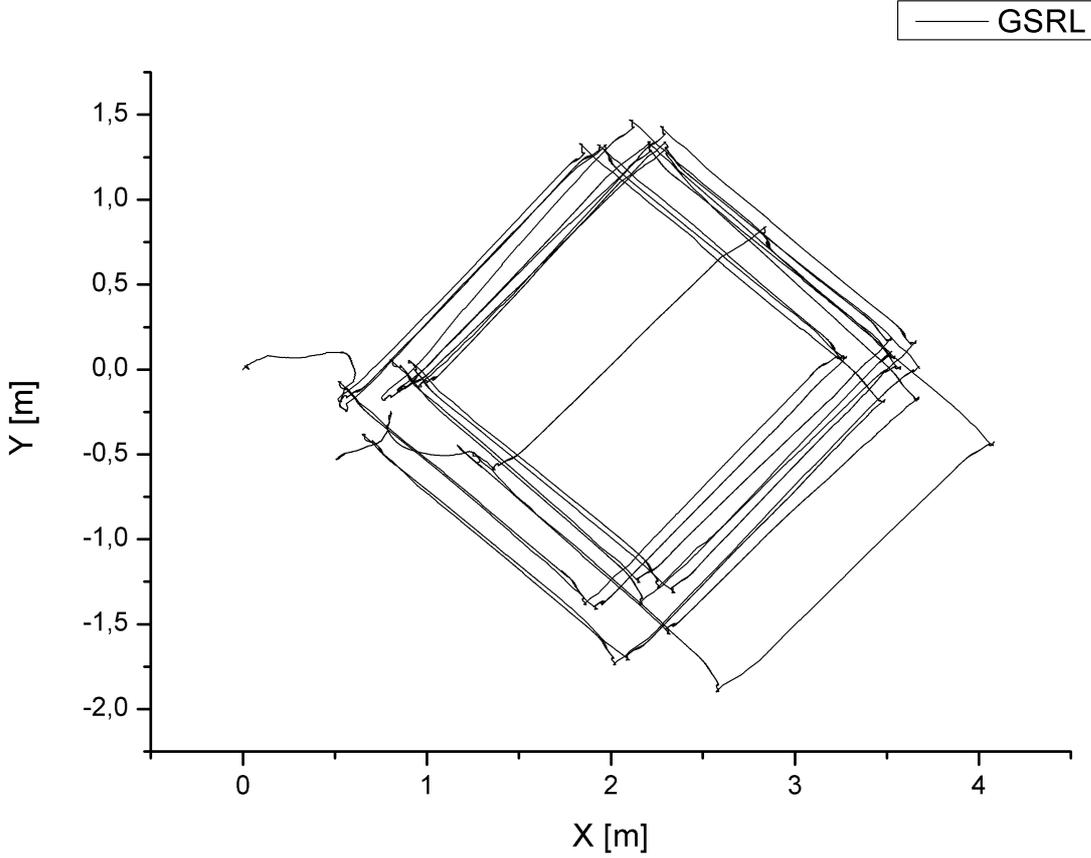


Figure 5.16: Position estimation by the GSRL for the sparse indoors validation dataset. Ten consecutive squares (2.5x2.5 m) with manually enforced start- and end positions. Due to the confined space of the test-tank, an obstacle-avoidance behavior was active at the same time, accounting for the offset of squares.

tank, starting at a defined position at the surface. The defined position was assured by manually holding the vehicle in a corner against the walls. The vehicle speed was selected to be 0.1 m s^{-1} . After ten trajectories were completed, the vehicle was manually returned to the defined starting position. The deviation of the start to end position was interpreted as deviation over the driven distance. The results were a deviation of $d_s = 0.7554 \text{ m}$ after a traveled distance of $|\eta| = 106.2 \text{ m}$, resulting in a relative deviation of $d_{r,s} = 0.71 \%$. The resulting x/y position as well as the x/y/z position are shown in figure 5.17.

It needs to be noted that the trajectory follower used in this experiment was complemented by an obstacle-avoidance module, which used the scanning sonar to prevent the

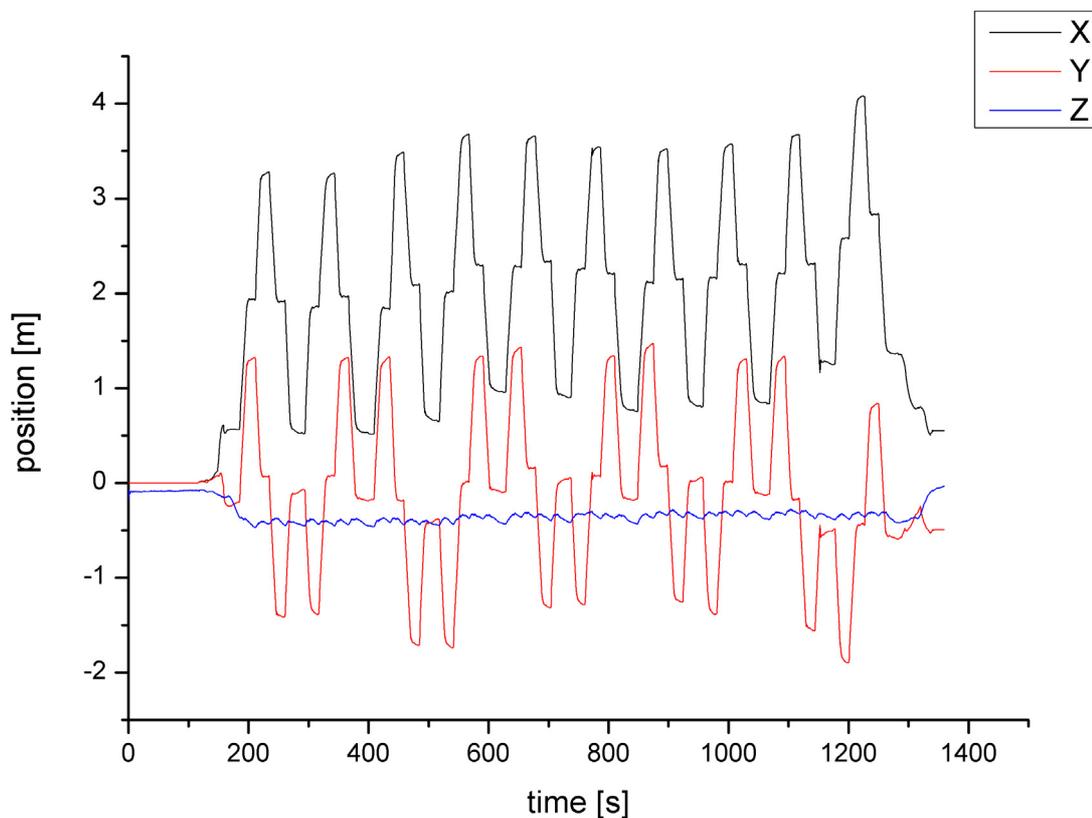


Figure 5.17: X, Y, and Z components of the position estimation of the GSRL during the sparse indoors validation dataset.

vehicle from colliding with obstacles. This resulted in a slight offset of the trajectories as seen in figure 5.16.

5.2.7 Sparse Validation of Reference Measurement (Unisee)

The second experiment was conducted in the Unisee environment with a similar setup, but with longer trajectories under more realistic circumstances. Again the vehicle's start and end position were fixed (holding it against the pier), but this time it performed one single large rectangle of 100x100 m. The vehicle speed was selected to be 0.3 m s^{-1} . The result was a deviation of $d_s = 2.2049 \text{ m}$ after a traveled distance of $|\eta| = 542.2 \text{ m}$, resulting in a relative deviation of $d_{r,s} = 0.41 \%$. The resulting trajectory and its manual fixing positions are shown in figure 5.18.

The fact that the second experiment resulted in a better deviation, despite being

5. EXPERIMENTS

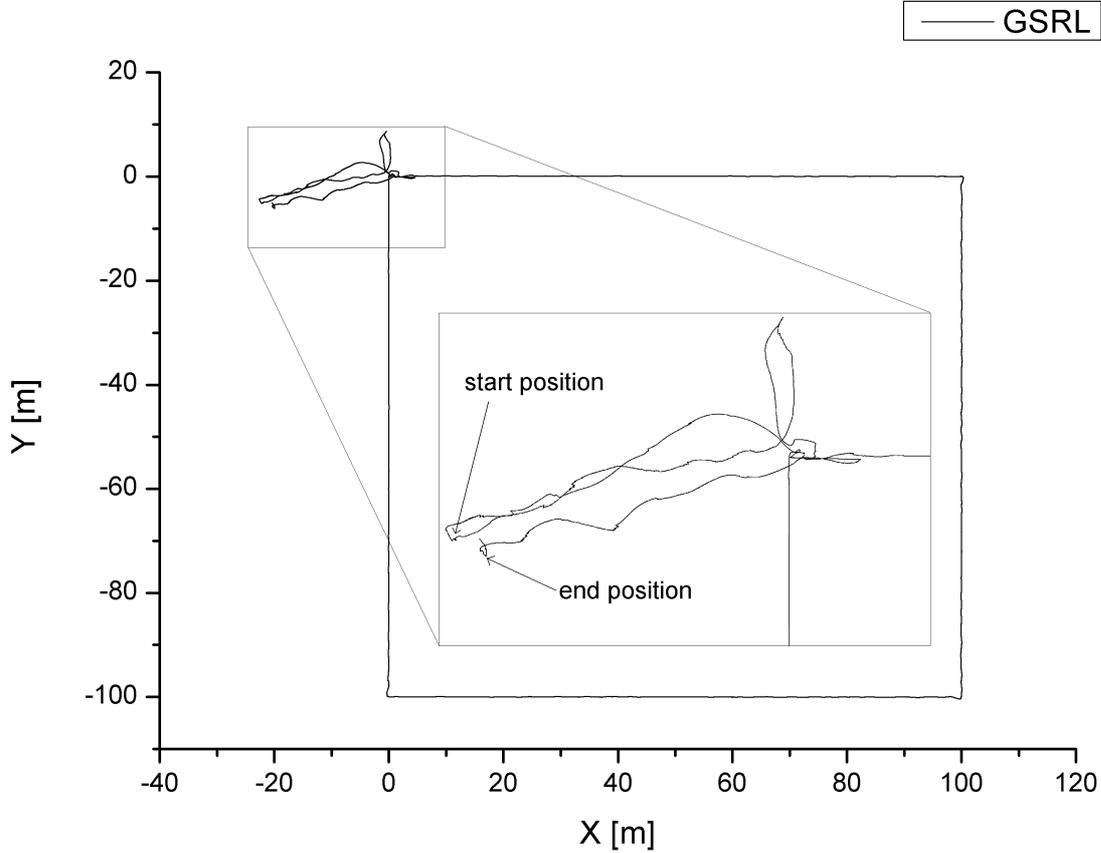


Figure 5.18: Position estimation by the GSRL for the sparse outdoors validation dataset. One large square (100x100 m) with manually enforced start- and end positions.

executed under more realistic conditions is explained by the smaller number of corners in the driven trajectory. Since the vehicle stops at each corner before it executes the turn, it experiences more pitch and roll. This increased the accumulated error of the DVL and thus resulting in worse results.

5.2.8 Indoors Validation (Glass Tank)

The indoors dataset was used to show the correctness of the validation approach used in this thesis since two measurements could be used here: the normal comparison measurement using the GSRL, as well as the manual reference measurement similar to section 5.2.6. The indoors dataset was recorded in clear water conditions, without any special lighting conditions, with a distance to the ground of 1.5 m and an average speed

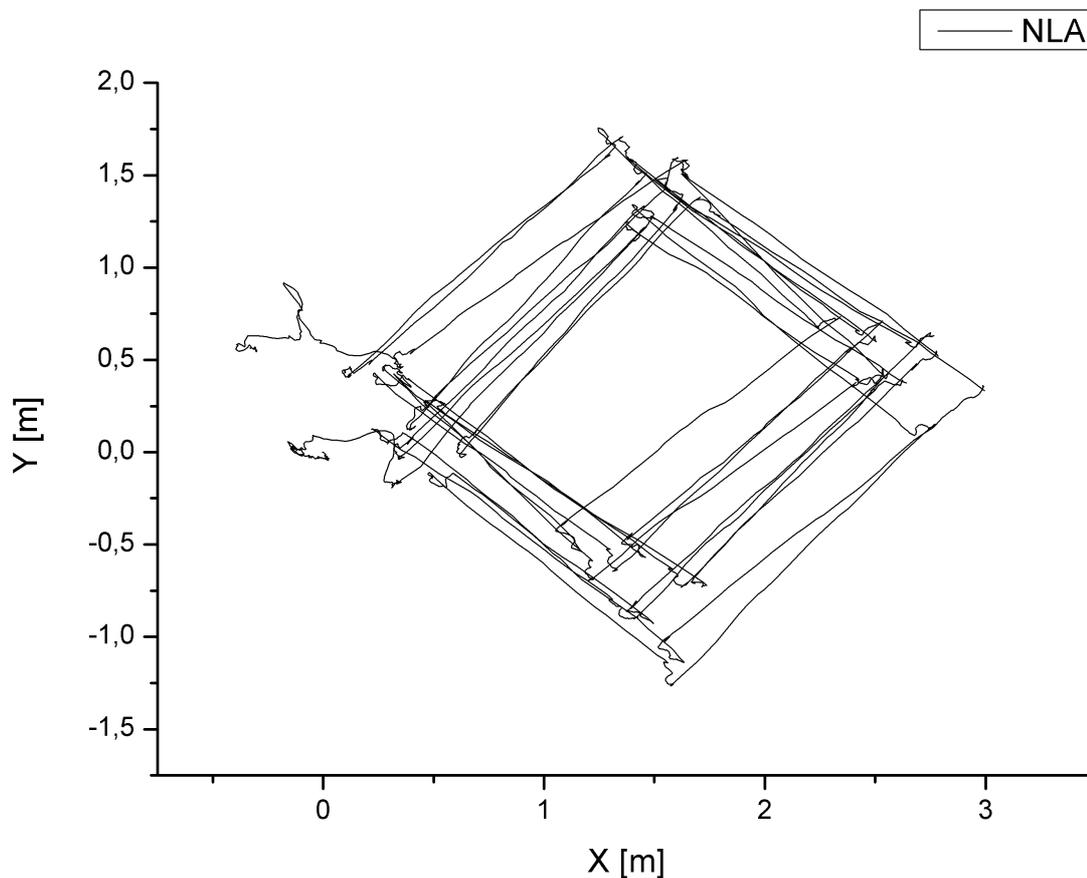


Figure 5.19: Trajectories as estimated by the SURE-SLAM for the indoors validation dataset.

of 0.1 m s^{-1} . Again ten square trajectories were driven with a $2.5 \times 2.5 \text{ m}$ size. Start- and stop position were manually defined as before. The resulting trajectory as estimated by the NLA is shown in figure 5.19. The sparse measurement resulted in a deviation of $d_s = 0.7855 \text{ m}$, after a traveled distance of $|\eta| = 103.9 \text{ m}$. A relative deviation of $d_{r,s} = 0.76 \%$ was calculated. These values are very close to their GSRL counterparts and thus showing that SURE-SLAM is working as designed.

Comparing the estimated trajectories of SURE-SLAM and the GSRL, the position difference at the end of the trajectory was $d = 0.236 \text{ m}$, resulting in a relative deviation of $d_r = 0.22 \%$. The standard deviation of the error between the two estimated trajectories was $\sigma = 0.081224 \text{ m}$. The fact that these values are significantly lower than in the sparse case underlines the necessity of considering the deviation of the reference

5. EXPERIMENTS

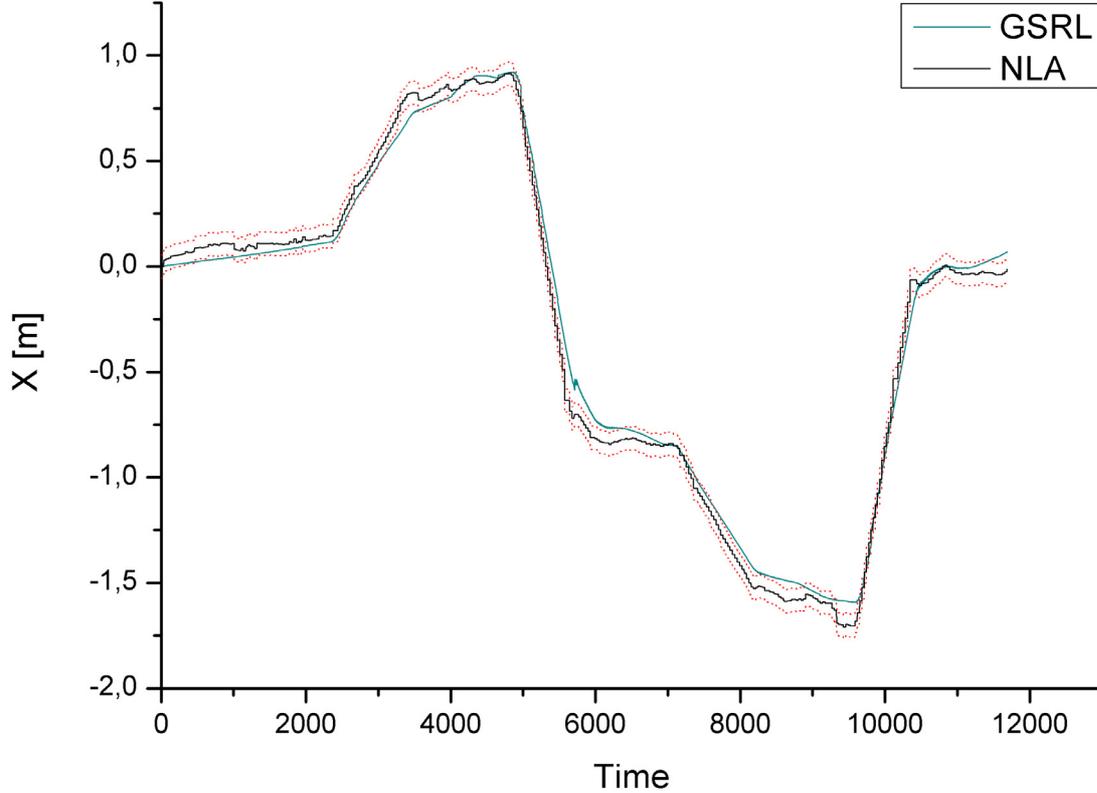


Figure 5.20: X-component of the first loop of the trajectory with standard deviation of GSRL and NLA of the indoors validation dataset.

localization when computing such quantities.

Since this experiment was conducted in a semi-controlled environment (the glass basin) it was partially repeatable. While the environment was untouched (water quality, ground) the exact vehicle trajectory could not be reproduced. Nevertheless, the experiment was repeated twice, DAGON following the same pre-set trajectory. The results were very close to each other, the values presented above are the values of the worst run. The other two runs had standard deviations of $\sigma = 0.05482$ m and $\sigma = 0.07938$ m respectively. A mosaic image created from the images, poses and correspondences is shown in figure 5.33.

5.2.9 Dynamic Environment (Kiel)

After most of the experiments were conducted in calm water, the effect of moving water and moving ground objects on the algorithm's performance was to be tested. The trials

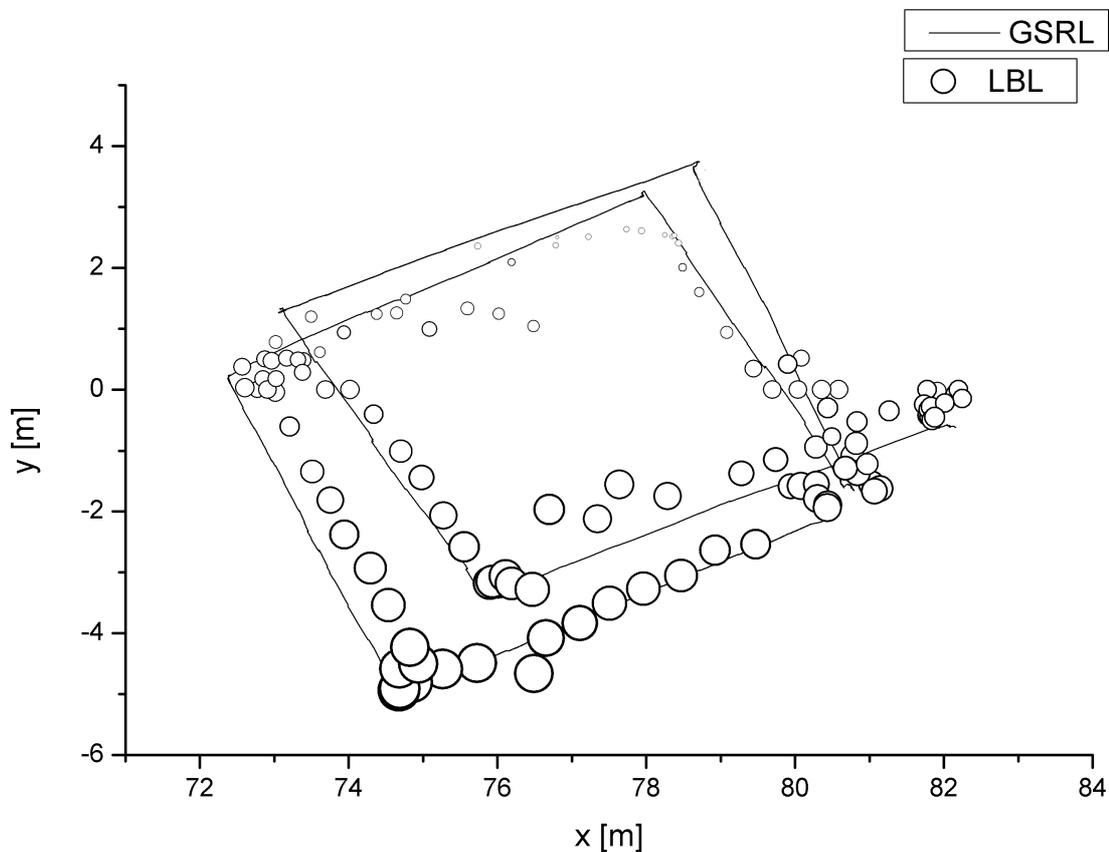


Figure 5.21: Estimated trajectory with LBL readings in the Kiel harbor (fall 2011).

in Kiel were the first occasion where such data could be recorded. Since the weather was rough, there was a lot of current and waves in the water. Due to its shallowness the harbor had plant growth at the bottom, which moved together with the water. This can potentially be a problem for the algorithm since it has to determine which of the image portions belong to the ground (being useful for localization) and which are motion of plants or ground objects (having to be treated as noise). During the 4-day trials a number of individual experiments were conducted, most of them discussed in section 6.1 on failure cases. This series of test was of preparatory nature to gain the operational experience for the 2012 outdoor test campaign in Rostock.

After the evaluation of the Rostock-trials in fall 2012, the failure of the LBL system during these trials became clear (for details see section 5.1.8.1). This made the Kiel trials the only sea-trials with a working LBL system. Unfortunately, no validation experiments were conducted in Kiel, so no sparse validation data is available for the

5. EXPERIMENTS

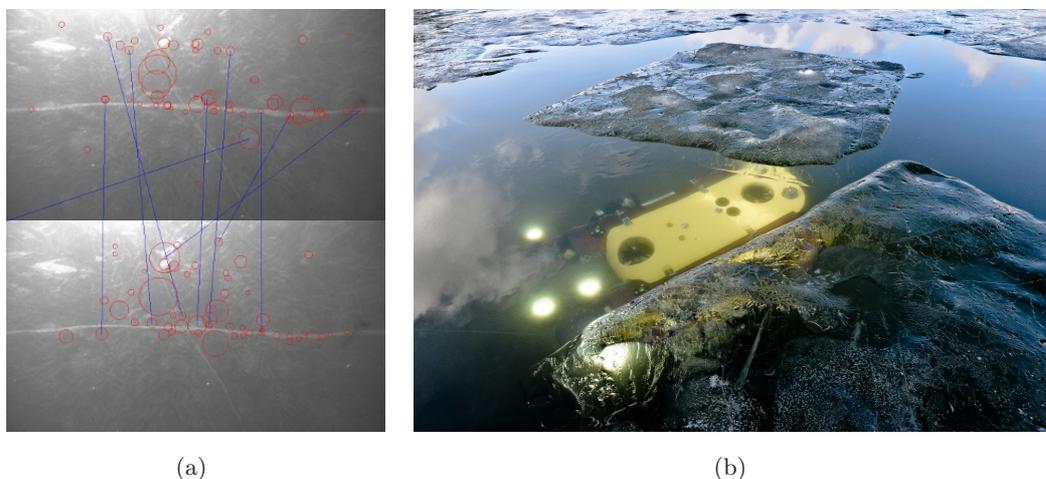


Figure 5.22: Under-ice trials winter 2011/12. Sample frame of under-ice dataset with unsuccessful inter-frame correspondences due to non-Lambertian surface (a). The vehicle surfacing after a successful trajectory (b).

individual experiments (the AUV's start- and end-location were not manually enforced or otherwise recorded). Nevertheless, the principle for a LBL-stabilized could be shown. Similar to the basin experiments square trajectories with an edge lengths of 6 m were driven. The LBL system was deployed at the extreme positions of the pier, covering an area of 60x80 m. The resulting trajectories and LBL readings are shown in figure 5.21.

5.2.10 Under Ice (Unisee)

In Winter 2011/2012 a unique opportunity arose as the test lake froze completely due to the low temperatures. The Bremen fire-department was kind enough to cut a hole into the 0.2 m thick ice, which could be used to lower the AUV and explore the lake under ice. A special challenge for the vehicle was the task to return to this hole after the mission – otherwise it would have been stuck. During this field-test the cameras were tilted upwards, and the vision algorithm attempted to localize using the ice surface. This failed with all selected illumination techniques (no vehicle illumination, just ambient illumination from sun, vehicle illumination together with ambient illumination, vehicle illumination only at night). The reason for this are the reflective properties of ice. Ice is a non-Lambertian reflector and changes its visual appearance with the angle of view. This makes correspondence estimation nearly impossible and thus the algorithm

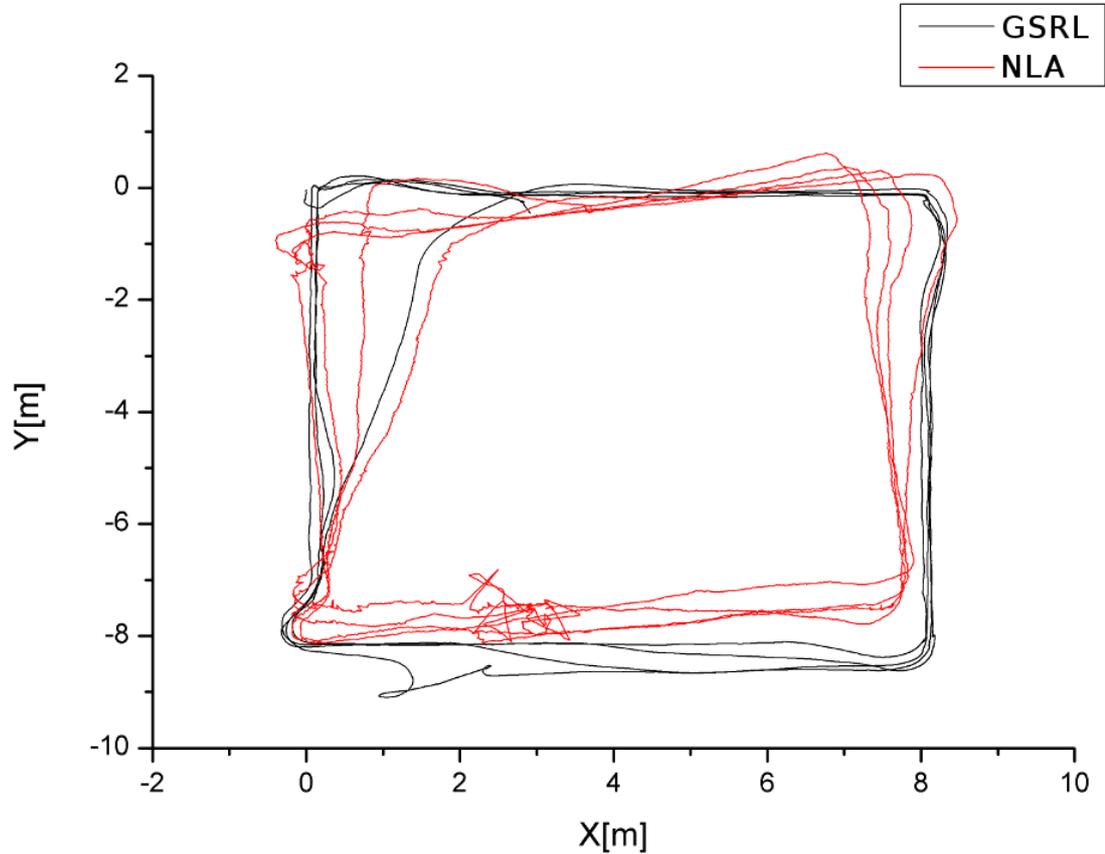


Figure 5.23: Trajectories as estimated by the GSRL and SURE-SLAM for the outdoors validation dataset.

fails to localize the vehicle, making this one of the failure cases. A sample frame of the under-ice dataset with unsuccessful inter-frame correspondences is shown in figure 5.22.

5.2.11 Outdoors Validation (Rostock)

The outdoors dataset was recorded in open water (Baltic Sea 2012), with moderately turbid water (≈ 3 FTU), sunny outdoors lighting conditions, with a distance to the ground of 2.5 m and an average vehicle speed of 0.3 m s^{-1} . There was a steady current of about 0.4 m s^{-1} , and about 0.5 m of waves. Five 8×8 m rectangular trajectories were driven. The resulting trajectories as estimated by the GSRL and NLA are shown in figure 5.23. The position difference at the end of the trajectory was $d = 0.712 \text{ m}$, with a

5. EXPERIMENTS

driven distance of $|\eta| = 160$ m, resulting in a relative deviation of $d_r = 0.45\%$. The standard deviation of the error between the two estimated trajectories was $\sigma = 0.15298$ m.

The number of nodes in the graph was $|V| = 2464$. The length of the path from the start node to the final node was $|p(V_0, V_n)| = 241$, the longest path in the graph had a length of 438 nodes. The factors of about 10 (respective 5) between these numbers show the impact of loop closing, especially if the resulting effective deviations are considered: applying the relative deviation backwards on the reduced length of $p(V_0, V_n)$, the resulting absolute deviation at the end was $d' = 0.04$ m, since the distance traveled along that path was only 8.8 m in the graph. This is the point where the missing LBL data would have improved the interpretation of the data, comparing SURE-SLAM to a drift-compensated GSRL after a 45 minute mission.

5.2.12 Brute-Force Loop-Closing (Rostock)

The aim of the brute-force loop-closing experiment was to determine if the feature-based approach for loop-close detection was robust to false-positives and wrong-negatives. For this purpose a special trajectory was driven during the Rostock trials: three squares with 8 m side length, with an offset of half the side length in both the x and y axes. This resulted in a cascade of three squares shown in figure 5.24. The idea of this trajectory was to create a number of areas where loop-closing was possible, even with rotated orientations. The resulting 1500 image pairs were then exhaustively matched with each other to determine if the algorithm would find loop closings. This required 1.2 million tests and lasted 18 h. A false-positive was defined as the detection of a loop-close where there was none, and a wrong-negative if a true loop-close was omitted. Initially the algorithm found a number of false-positives, which could be remedied by increasing the number of required feature-matches to 8 (theoretically 5 are sufficient, but 8 are the numerically safer next best category (26)). After this modification, no false-positives were found. All possible loop-closing positions but one were found, yielding only one wrong-negative measurement. After investigation this could be remedied as well: The position which was not detected for loop closing was the intersection of the first and the last square. Manual analysis of the data showed that while the trajectory should have overlapped at this point, due to drift it really did not, making a loop-close impossible. As conclusion for this experiment, the feasibility of the feature-based approach could be validated on a real-world dataset. The complexity of this task can be seen by the

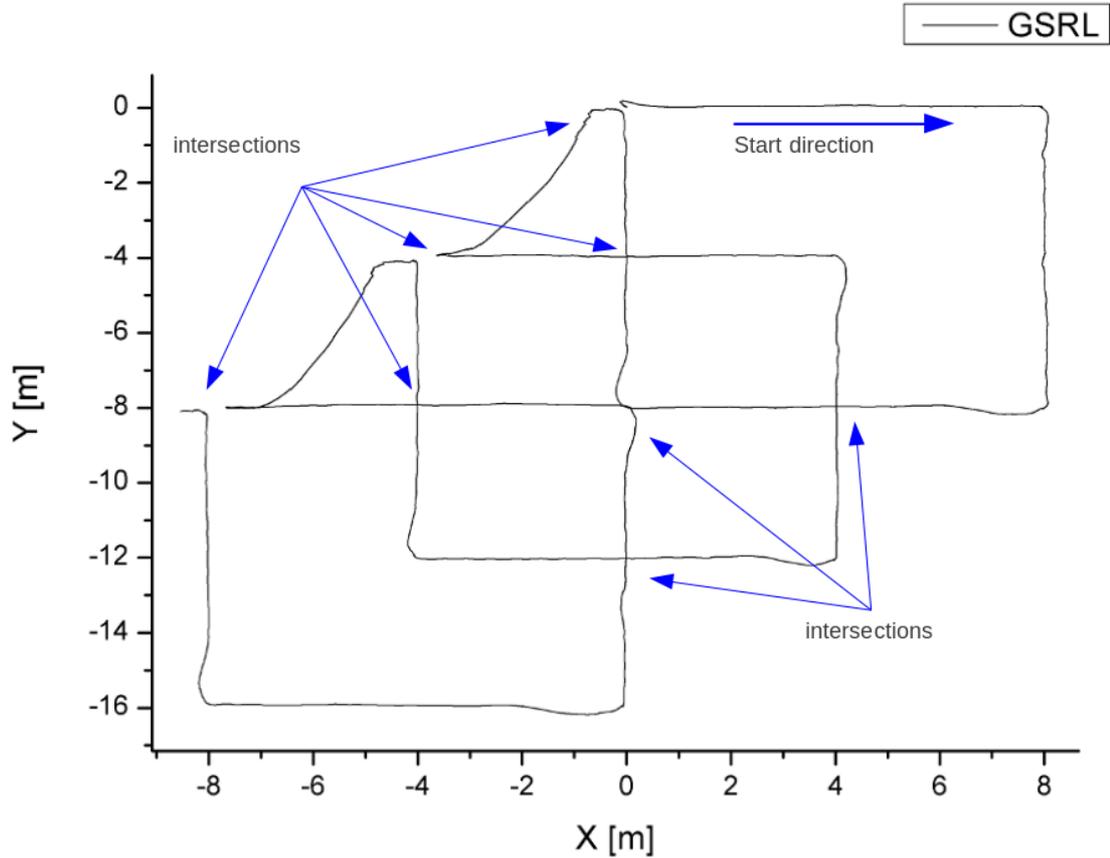


Figure 5.24: The trajectory of the brute-force loop-closing experiment with loop-closure locations.

two sample image pairs shown in figure 5.25. For a human to determine if the images contain (possibly rotated) overlap is a very tedious task, since there is a high degree of self-similarity and very limited global structure in the images.

5.2.13 Long-Term Stability (Glass Tank)

In order to test the long-term stability of the algorithm, an experiment using DAGON's station-keeping function was conducted in the glass tank. After initialization and manual enforcement of the starting position in one corner of the tank (similar to section 5.2.6), the vehicle was moved to the center of the tank. There the station-keeping was activated and the vehicle kept hovering in the same place for 90 min. At the end, the vehicle was again returned to its starting point and the final position was man-

5. EXPERIMENTS

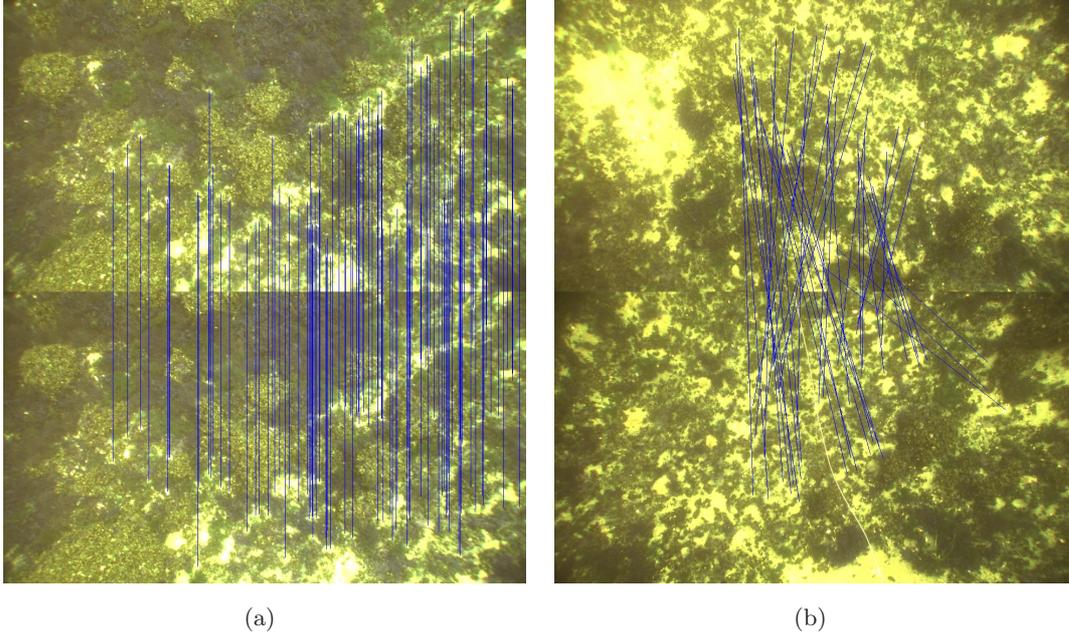


Figure 5.25: Visual data from the Rostock datasets. Inter-frame-matching of image-features (a). Loop-closing match of image features with a time-delay of 30 minutes (b).

ually corrected. Both the reference localization and SURE-SLAM localization were recorded. Figure 5.26 shows the estimated trajectory. The results were a deviation of $d_s = 0.2315$ m after a traveled distance of $|\eta| = 11.3$ m, resulting in a relative deviation of $d_r = 2.05\%$ measured by the GSRL. The orientation error was 2° . The SURE-SLAM measurement resulted in a $d_s = 0.0172$ m after a traveled distance of $|\eta| = 5.2$ m, resulting in a relative deviation of $d_r = 0.332\%$. No orientation error was present in the SURE-SLAM measurement. The shorter $|\eta|$ measurement for SURE-SLAM stems from the fact that since during position-keeping constant loop-closures can be obtained, the maximum length of the graph is very low. For a completely dead-reckoning based approach (GSRL) there is no distinction between hovering and driving, resulting in an accumulation of small motion to a longer overall path.

5.2.14 Sparse Environment (Synthetic)

In this experiment the required amount of texture information for stable operation of the algorithm was to be tested. For this purpose the synthetic dataset was modified so a varying amount of texture could be used. The visible texture of the floor in the

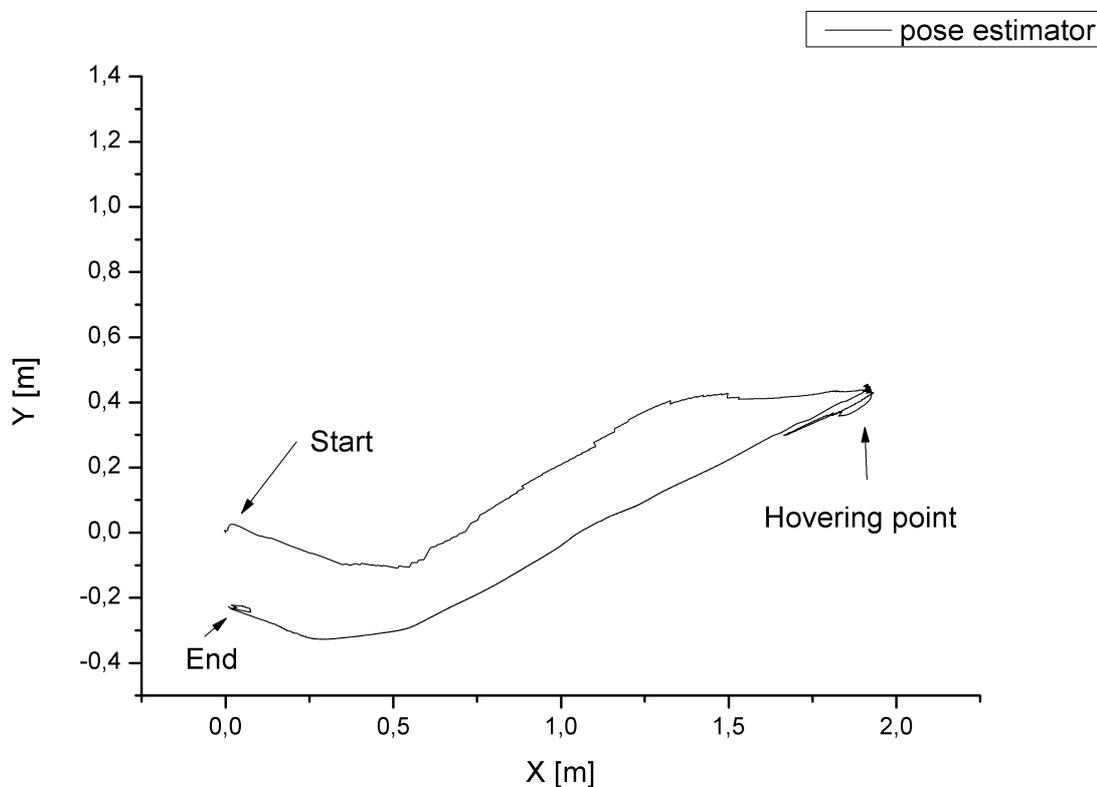


Figure 5.26: Trajectory as estimated by the GSRL for the long-term hovering experiment. After starting in one corner of the glass tank the vehicle was hovering for 90 min in the center, and then returned to the initial position.

synthetic dataset is a combination of the roughness of the bottom mesh (its “ruggedness”) and the detail of its diffuse texture. Together with the vehicles lights, a three dimensional structure and resulting texture is observed by the cameras. The roughness of the surface is not directly modeled as displacement on the mesh level, since experiments with such displacement required the mesh resolution to be too fine for reasonable rendering times. Instead, a parametric texture is used on the “bump”-channel of the used material, resulting in a per-pixel z-displacement of the surface. Unfortunately, the resulting absolute z-displacement is not directly measurable. The textures used for both the bump map and the diffuse map are based on fractals in order to provide easy parameterization and assured non-repetivity. For both textures three different parameter sets were used to create a total of nine scenes for evaluation. In all nine scenes the same trajectory for the virtual AUV was rendered, resulting in 250 stereo pairs. The

5. EXPERIMENTS

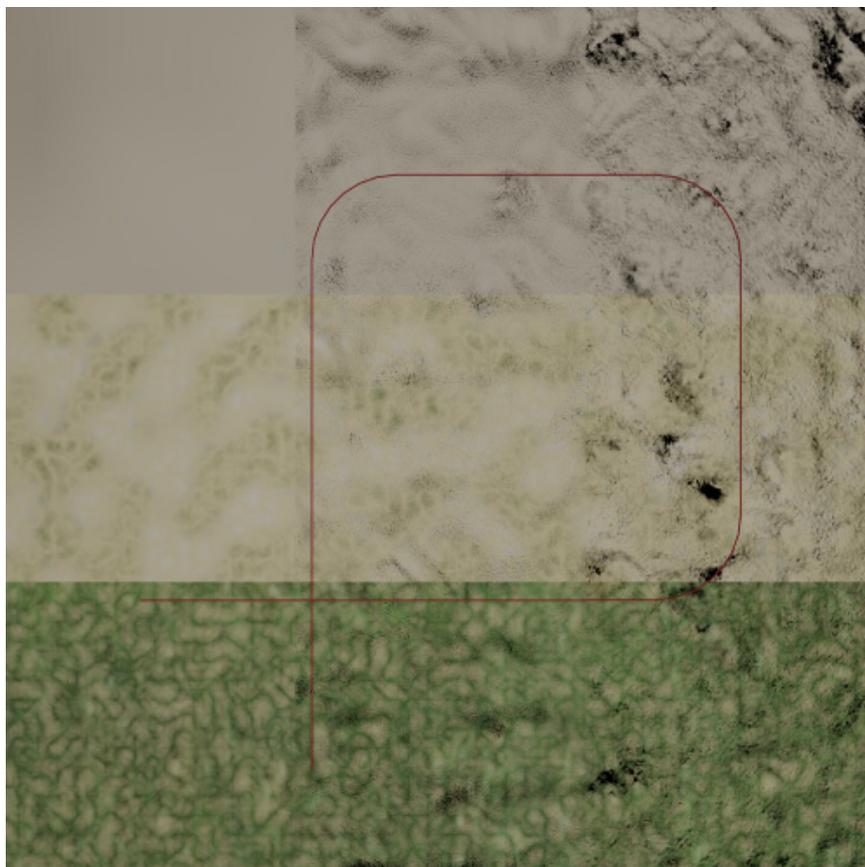


Figure 5.27: A combination of the nine synthetic scenes used for testing in a variably sparse environment. The vehicle trajectory is shown in red.

trajectory was a square of 5 m edge length with overlap near the start/end position in order to evaluate loop closing capability. A combined picture of all nine scenes is shown in figure 5.27.

Two tests were conducted on these datasets: performance of visual odometry and loop closing. The results are summarized in table 5.2. While visual odometry works in all but the low ruggedness/low texture and low ruggedness/medium texture, loop closing shows a more diverse picture: While it worked in all high-texture cases, it only worked well in the high ruggedness/medium texture case, did only barely work in the medium ruggedness/medium texture and medium ruggedness/high texture case (with only one resp. two detected loop closures) and did not work in the medium ruggedness/medium texture case. Two reconstructed trajectories are shown in figure

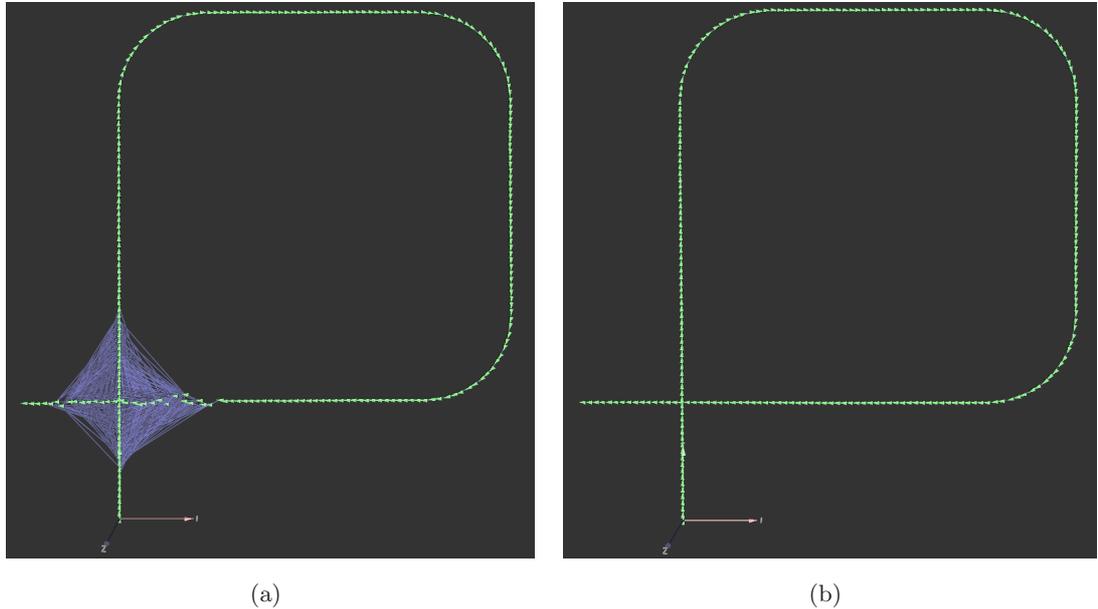


Figure 5.28: Reconstructed trajectories from the sparse synthetic datasets. Complete set of loop closures in the low ruggedness/high texture dataset (a). No detected loop closures but working visual odometry in the medium ruggedness/medium texture dataset (b).

5.28.

5.2.15 Impact of Internal Navigation Hardware on NLA quality (Glass Basin)

As described in section 4, the usage of intrinsic vehicle sensors (INH - internal navigation sensors) is not a necessity for the algorithm's operation. If this sensor data is available, it can improve the robustness of the SURE-SLAM significantly. The usage of INH can counter three problems that may arise with a purely-vision solution:

- Camera calibration issues

The visual algorithm relies on intrinsic and extrinsic camera calibration. If the calibration is broken, a direct effect can be seen on data quality. Since the vehicle is operating in real environments, slight changes in the calibration happen quickly (e.g. mis-alignment of the rotational component between the two cameras).

- Fast rotations

Rotations can induce significant angular motion of features in the camera images.

5. EXPERIMENTS

Table 5.2: Results of the sparse environment experiments. The first value in each cell is the feasibility of the terrain for visual odometry, the second value the number of detected loop closures.

	low ruggedness	medium ruggedness	high ruggedness
low texture	no/0	yes/1	yes/2
medium texture	no/0	yes/0	yes/2
high texture	yes/89	yes/67	yes/41

This makes correspondence detection harder and reduces the resulting measurement quality.

- Limited depth resolution

The resolution of the depth perception (z component of η_1) is significantly lower than the x and y components. As stated in section 3.3.7 at 3 m operating distance the cameras have a pixel resolution of 2 mm or 4 mm for the reduced resolution used in online processing. With a stereo baseline of 30 cm the depth resolution is only 3.8 cm and thus an order of magnitude worse. This directly results in a reduced reconstruction quality of the z component.

The experimental setup was similar to the sparse indoor validation described in section 5.2.6. The vehicle was driving 10 complete rounds, while the NLA measurement was computed with and without INH aiding. The speed of the turning behavior was doubled in the trajectory follower, and the camera calibration artificially corrupted by rotating the left camera head slightly. The resulting estimated trajectories are shown in figures 5.29,5.30. It can clearly be seen that the small errors induced at the corners of the trajectory result in an overall orientation drift of more than 45° after 10 rounds, significantly reducing the chance for graph-based loop-closing. This effect was artificially exaggerated by corrupting camera calibration and increasing vehicle speed in this experiment. It can be expected that during carefully planned operation the penalty on accuracy would be less severe.

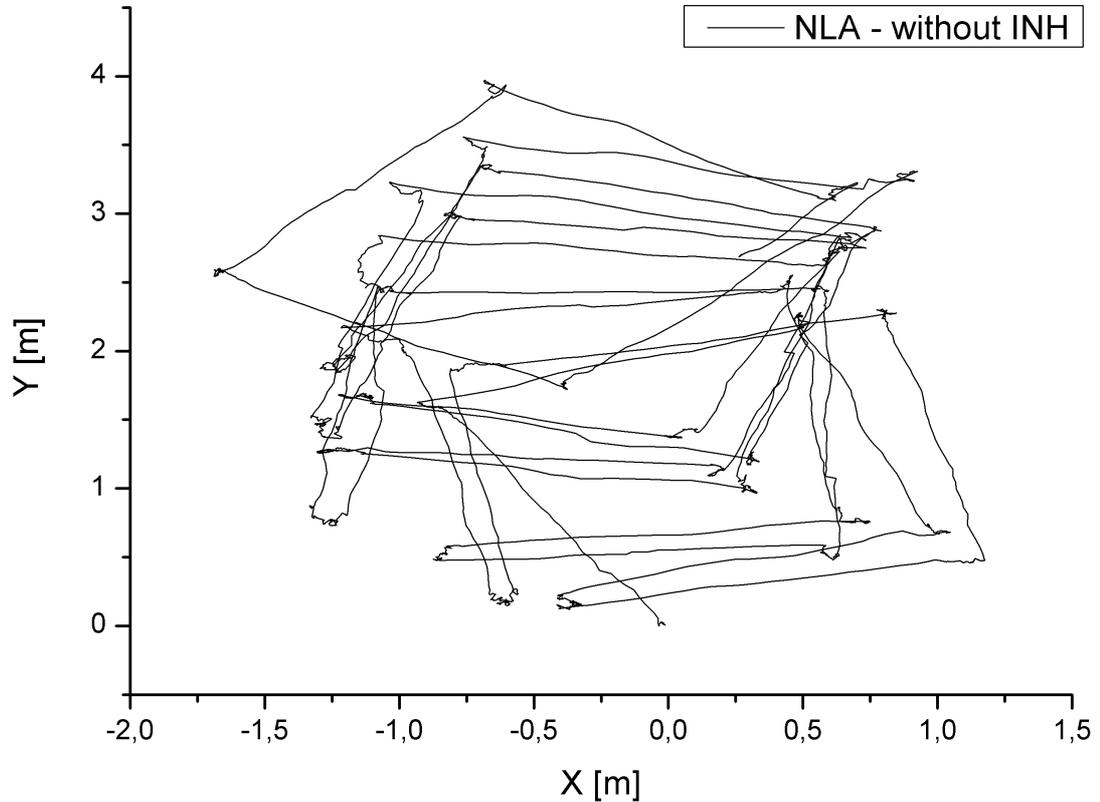


Figure 5.29: Impact of INH (DPS, AHRS and FOG) on NLA quality with corrupted camera calibration and increased vehicle speed. INH de-activated, visual odometry only.

5.2.16 SLAM vs. Visual Odometry (Unisee)

In this experiment the impact of SLAM in comparison with visual odometry was to be examined. For this purpose a similar setup as in section 5.2.7 was used, and a 8x8 m trajectory performed at the steep shore of the lake. Instead of running the NLA and GSRL in parallel, the visual odometry and the SURE-SLAM algorithms were running in competition this time, and their sparse deviations were compared afterwards. The sparse deviation difference between the two measurements was 0.808 847 m, the resulting 3d plot is shown in figure 5.31. The gap in the visual odometry only measurement is well visible, since the visual odometry was run without intrinsic vehicle sensors its z-component error was significant. Loop closing in the SURE-SLAM measurement remedies this problem by optimization of the entire graph and the resulting correction of the z-component error.

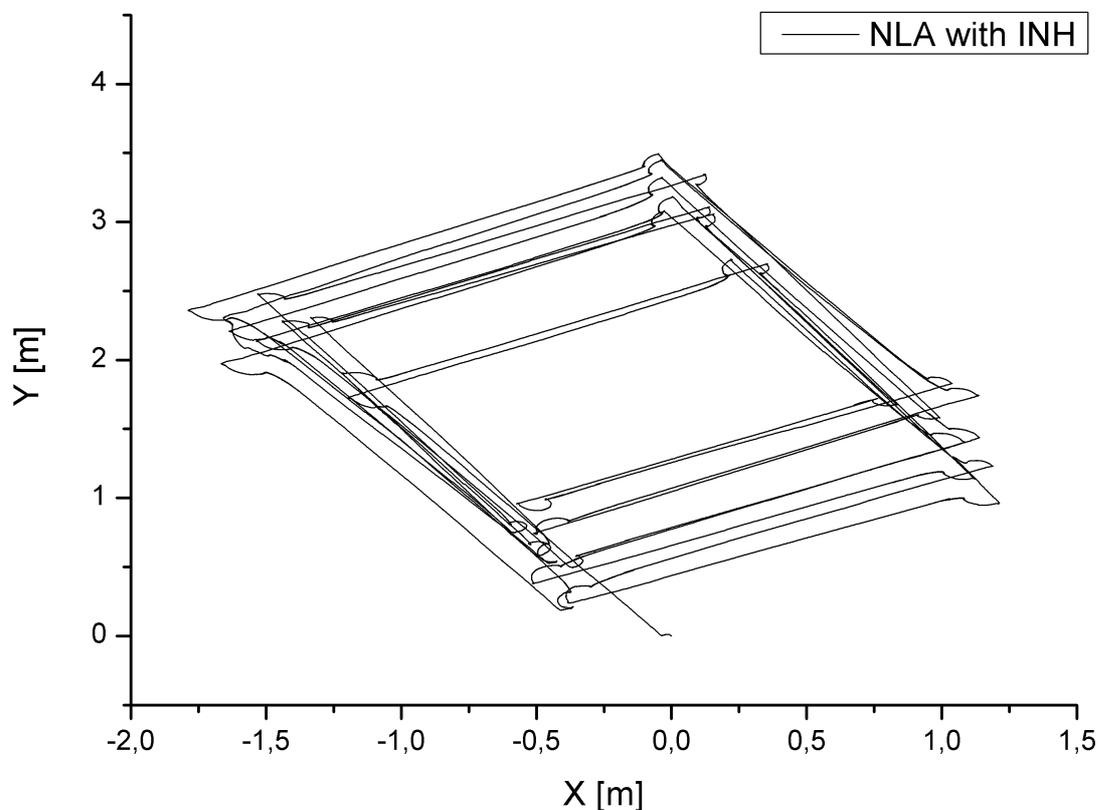


Figure 5.30: Impact of INH (DPS, AHRS and FOG) on NLA quality with corrupted camera calibration and increased vehicle speed. IHN activated, orientation drift compensated.

5.2.17 Sherpa (Space Exploration Hall)

The algorithm was not only tested on underwater vehicles, but on land-based vehicles as well. The most prominent among them was the robot “Sherpa”, a four-legged walking/rolling robot developed for space-exploration in the RimRes¹ project. The robot is equipped with a stereo camera system which can be swiveled to face to the floor (see figure 5.32(a)). In order to test the feasibility of the algorithm in this different scenario, a number of experiments were conducted. While due to time constraints no complete analysis could be made, it could be validated that the visual odometry portion of the algorithm was working fine (see figure 5.32(b)).

A special problem in this scenario was the presence of a moving shadow induced

¹<http://robotik.dfki-bremen.de/de/forschung/projekte/rimres.html>

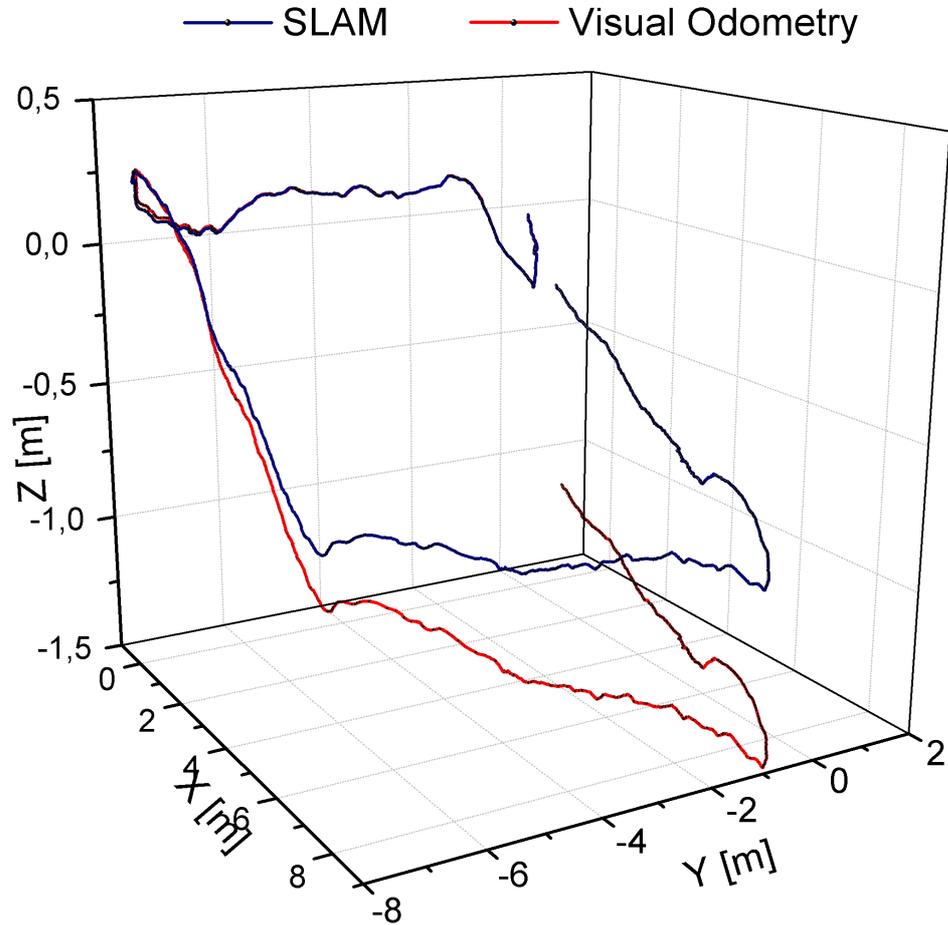


Figure 5.31: Impact of loop closing on trajectory: only visual odometry has a large gap in the trajectory, the SLAM trajectory has closed that gap by loop closing.

by the robot. This resulted from a static external light source (representing the sun) and a moving robot. The effect on the algorithm was a non-uniform illumination of the images, and thus a resulting omission of features in the less-illuminated regions. This can be remedied by methods such as histogram equalization, which have to be included in the pre-processing-step.

5.2.18 Mosaicking (Glass Basin)

One of the shortcomings of the SURE-SLAM algorithm is its failure to intrinsically create human-readable maps as output. As a graph-based SLAM approach all map data is located in the graph nodes. This is perfectly suitable as map for the robot,

5. EXPERIMENTS

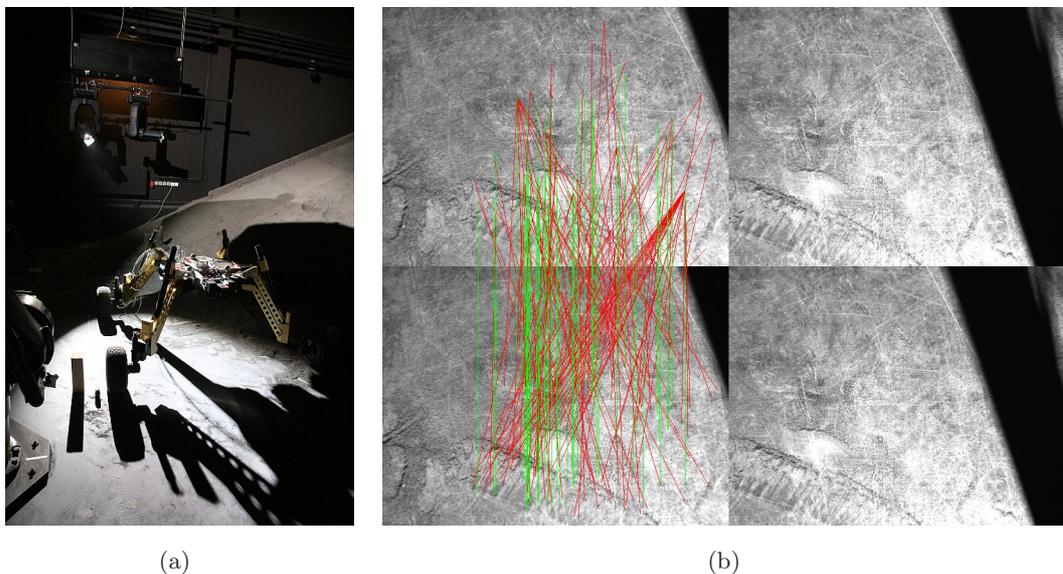


Figure 5.32: The Sherpa robot with its stereo camera system surveying the ground (a). Geometric relations of two consecutive stereo pairs with features recorded by the Sherpa robot (b).

which uses it for localization and modifies it according to new readings (e.g. loop-closing). While it is possible to use the graph-based map as basis for a human-readable map (e.g. a mosaic or a 3d reconstruction), this requires a lot of additional work for high-quality results. The PhD-thesis by Johnson-Roberson (31) (more details on it were given in section 1.2.2.3) shows exactly such an export component. Nevertheless, as proof-of-concept a mosaicking module was implemented for a simple 2d environment case. Here the ground is treated as planar, reducing the problem to an estimation of homographies and especially the infinite homography of all images (42). An example mosaic of the glass basin testing environment is shown in figure 5.33.

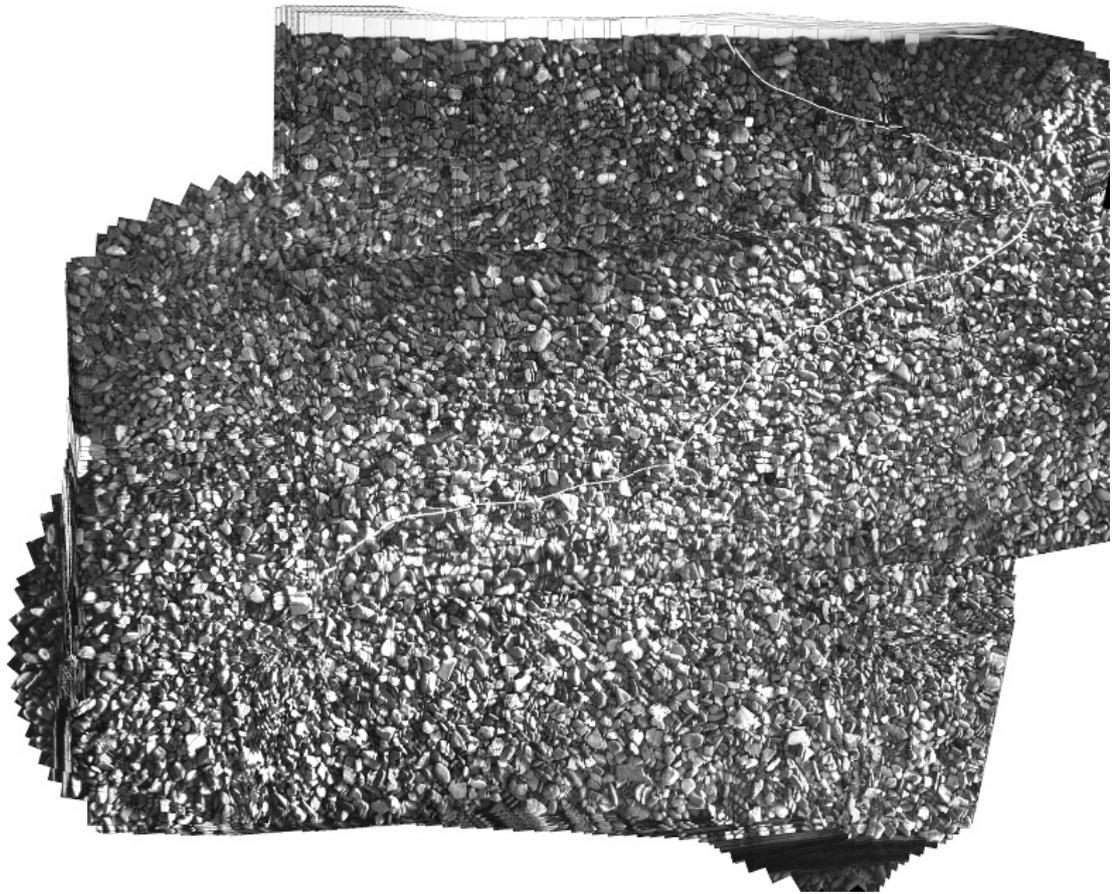


Figure 5.33: Mosaic created from images and correspondences of the glass basin environment.

5. EXPERIMENTS

6

Conclusion

A summary of the results is given in table 6.1. The results are discussed in detail in the following sections.

6.1 Description of Failure Cases

There are a number of circumstances where the described visual-SLAM based NLA fails. It is important to keep the operational conditions within these parameters if SURE-SLAM is to be used as primary localization and navigation solution:

- Distance to ground

It is a necessity for the visual system to yield usable results that the ground is visible and within the triangulation range of the stereo camera system. The optimal operation conditions are within 2 m to 5 m stand-off distance to the ground (for 30 cm stereo baseline), with operation still possible up to 1.2 m to 7.5 m. Further reduced stand-off distance results in too little stereo overlap, while greater distance reduces depth resolution, both resulting in poor localization results or even drop-outs.

- Turbidity

Water turbidity acts as an image blurring agent and results in reduced performance of the feature extractor. The system has been tested at a stand-off distance of 2 m with a turbidity range of 0-10 FTU. Turbidity greater than 7 FTU resulted in decreased performance or complete drop-out.

6. CONCLUSION

Table 6.1: Performance results

experiment	$ \eta $	$d_{r,s}$	d_r	σ
sparse validation GSRL indoors	106.2 m	0.71 %	N/A	N/A
sparse validation GSRL outdoors	542.2 m	0.41 %	N/A	N/A
sparse long-term stability GSRL	11.3 m	2.05 %	N/A	N/A
sparse long-term stability SURE-SLAM	5.2 m	0.332 %	N/A	N/A
synthetic dataset SURE-SLAM	4500 m	0.1 %	0.1 %	3.317 m
validation SURE-SLAM indoors	103.9 m	0.76 %	0.22 %	0.081 224 m
validation SURE-SLAM outdoors	160.0 m	N/A	0.45 %	0.152 98 m
spider-cam trajectory	50.0 m	N/A	1.2 %	0.32 m

- Image overlap

The algorithm requires an image overlap of at least 35 %.

- Motion blur

Since motion blur greatly reduces the matching effectiveness of the feature detector, it has to be avoided by selection of appropriate illumination, exposure times and gain settings of the cameras.

- Repetitive structure

Overly repetitive structures lead to failure of the loop-closing capability of the algorithm. The two test cases for this were a swimming pool floor (tiled in two colors) and the net of a fish-farm in Norway. Both resulted in good results for local navigation, but loop-closing was limited to areas with additional structure (e.g. numbers on ground in the pool or anchor lines at the net).

- Non-Lambertian surfaces

The algorithm operates with the expectation of mainly diffusely reflecting surfaces (Lambertian scatterers). Specular reflecting surfaces such as ice change in appearance independently of their relief (height towards camera), which results in poor performance of feature matching.

- Overly dynamic ground

The algorithm will compute the motion relative to the motion of the ground. If

the ground is covered with dynamic structures (such as sea-grass) which move independently from the vehicle, the algorithm will report the more significant motion, i.e. if more than 50 % of the detected feature points are on moving sea-grass, the motion relative to the sea-grass will be computed. This is a direct effect of the RANSAC-filter used for the FM/homography filtering (for details see section 4.2.1.4).

6.2 Resulting Characteristics for GSRL

As conclusion of the two validation experiments conducted, the upper bound for the GSRL relative deviation value is $d_r(\text{GSRL}) = 0.75\%$. This value lies in the expected range for the DVL/AHRS combination as described above, and will be used for the remainder of this work.

It needs to be noted that the GSRL deviation is a composite value, since it combines position deviation and rotational deviation. The drift of the orientation estimation and the drift of the accumulated speed-over-ground measurements by the DVL both contribute to the final deviation. In order to estimate how much of the drift stems from orientation drift, the performance of the orientation estimator was evaluated separately. Using a long-term test where the vehicle was sitting outside of the water on a table, the angular random walk was recorded using the Allen-variance. A value of 3°h^{-1} was measured. Since this measurement was taken with no roll and pitch variation (which increase the error since only a single-axis FOG is used), the upper bound for random walk error in the orientation estimator was selected as 6°h^{-1} .

6.3 Resulting Characteristics for NLA

The resulting corrected deviations for the SURE-SLAM are $d_c = 1.033\text{ m}$ for the indoors dataset and $d_c = 1.912\text{ m}$ for the outdoors dataset after a traveled distance of $|\eta| = 106\text{ m}$ and $|\eta| = 160\text{ m}$ respectively, with corrected relative deviations of $d_{c,r} = 0.97\%$ and $d_{c,r} = 1.2\%$. These results lie in the expected range considering that no external reference was used in the measurements of the GSRL. They represent the upper bounds for the expected error of the SURE-SLAM as validated with the GSRL.

6. CONCLUSION

Table 6.2: Comparison between the SURE-SLAM and the GSRL. All values are upper limits/upper bounds.

	SURE-SLAM	GSRL
relative deviation	1.2 %	0.75 %
update frequency	8 Hz	12 Hz
robustness against noise	low	low
robustness against environmental conditions	medium (turbidity, distance)	medium (layering, structure)
computational complexity	high	low
impact on vehicle instrumentation	cameras, light	DVL, LBL
versatility	good	good

6.4 Comparison Between GSRL and NLA

As a result of all the experiments it can be summarized that SURE-SLAM performed well (see table 6.2). While its relative deviation is higher than of the GSRL, it is only slightly higher. The same can be said for its update frequency, which should be easily improvable with the advance of computation power on embedded systems. Most important is the robustness against environmental conditions. Here the strengths and weaknesses seem to be highly disjunct, showing the opportunity of combining both methods into an even better system (see section 6.5.1).

6.5 Future Work

This thesis has created a solid foundation for future work on underwater vehicle localization. A number of ideas for future work on this topic can already be mentioned. They can be separated into work with the algorithm and work with the vehicle DAGON. As a new location for larger-scale controlled-environment experiments the new underwater exploration hall will be available from the beginning of 2014. With its size of 23x19x8 m it allows significantly more space for testing. An impression of the exploration hall is given in figure 6.1.

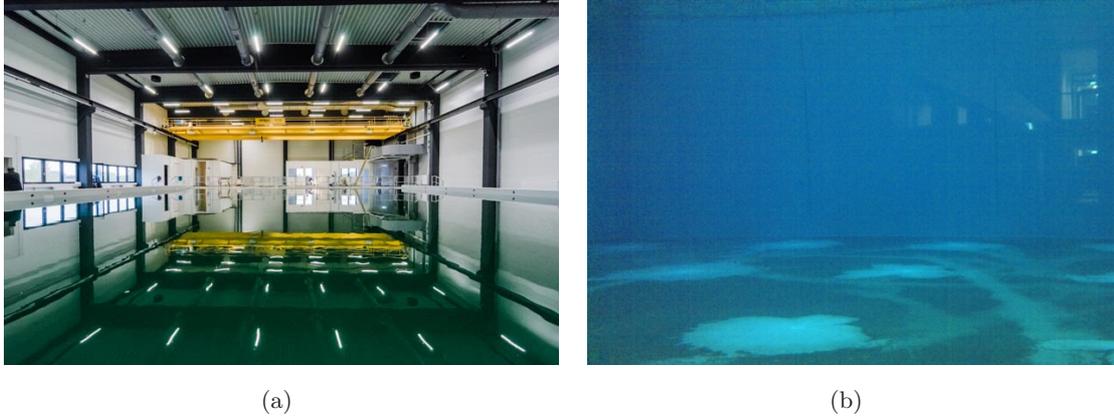


Figure 6.1: Topside view of the new exploration basin (a). View through the lower viewport into the basin (b).

6.5.1 Future of the SURE-SLAM Algorithm

Currently the SURE-SLAM algorithm is designed as standalone-localization solution for an underwater vehicle. If other modalities are available on a given vehicle, the data from these sensors can still be used to improve the overall localization quality. This could either be done in a post-processing filter, taking the estimates from all localization algorithms and fusing them, or more closely coupled. In the latter case techniques similar to the VAN approach by Eustice described in section 1.2.2.3 could be used. In the case of DAGON for example instead of using GSRL and SURE-SLAM in parallel competitively, they could work together to remedy their specific problems as summarized in table 6.2.

A limitation of the algorithm is its requirement for a stereo camera configuration which has its baseline perpendicular to the main motion axis. This combined with reasonable baseline lengths (30 cm on DAGON) makes the integration into typical torpedo-shaped AUVs complicated. One of the ideas to remedy this, is the usage of three calibrated cameras: two in the required stereo configuration with a small baseline (e.g. 10 cm) and a third camera perpendicular to this baseline with the originally required baseline. This should yield similar results with only slight modifications of the algorithm.

Both ideas will be put to test in the “Europa-Explorer” project, funded by the German Federal Ministry of Economics and Technology (BMWi). The project Europa-

6. CONCLUSION

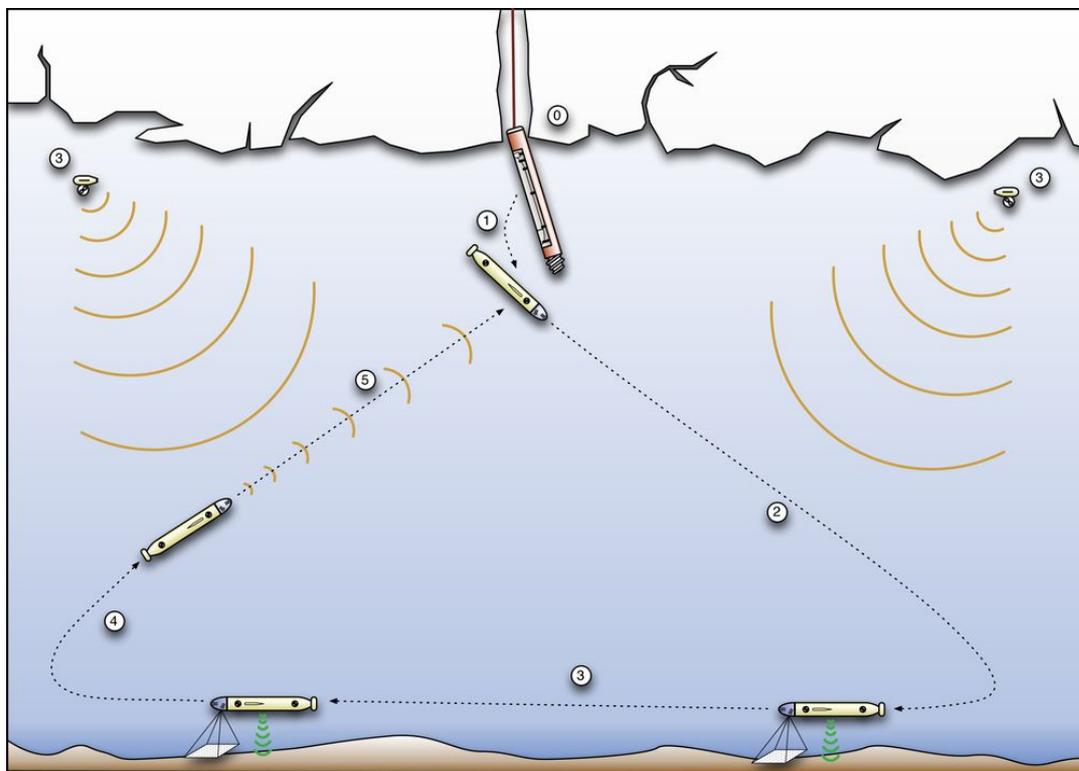


Figure 6.2: Schematic overview of a possible mission scenario for the Europa-Explorer project. 0) ice-drill penetrated the ice-shield. 1) AUV has been released from the payload compartment. 2) AUV descends to ocean floor. 3) exploration using cameras/sonar and internal sensors. 4) ascend to the ice/water boundary. 5) return to ice-drill (using autonomous localization buoys) and docking for energy/data exchange.

Explorer is a pilot survey for future missions to Jupiter's moon Europa. It focuses on the aspect of navigation of robotic systems on, and especially under the ice-shield of Europa. Below the surface an ocean comprised of liquid water is expected. After penetration of the ice-shield an exploration can be conducted. A possible mission scenario is drafted, which covers all aspects of an exploration from the time of landing until the transmission of the survey results (see figure 6.2). For this purpose a very small diameter vehicle is needed (around 20 cm) in order to fit through an autonomously drilled hole in the ice-shield. The navigation system of the vehicle will include the SURE-SLAM algorithm with the modifications mentioned above for ground-relative localization while surveying the ocean floor (27).

Finally the experiments including LBL measurements have to be done. As soon

as the device is repaired and was tested again in the university lake (or the new test basin), during the next scheduled outdoor mission (summer 2014 in Italy), the respective experiments will be executed.

6.5.2 Extensions of the SURE-SLAM Algorithm

During this work a number of ideas for extensions of the SURE-SLAM algorithm were collected and will be shortly presented in this section.

FPGA for Feature extraction Using a GPU as co-processor was only second choice when designing the vehicle and the algorithm. The ideal device for the extraction of features on a mobile robot is an FPGA, as was already stated in section 3.3.3. Since by now implementations of feature extraction on FPGAs became available (e.g. (59)), it is only a logical step to utilize them, making the vehicle benefit from a lower-powered co-processor and maybe even faster processing.

Loop-Closing Improvement In contrast to particle-filter-based or EKF-based SLAM approaches, graph-SLAM approaches have no problem with so called “delayed states” (29). The process of loop close integration into the pose graph is de-coupled from the data acquisition. This means if at any given time the available processing power does not permit complete evaluation of all loop closing candidates (see 4.2.3), this work can be delayed to a time where more processing power is available. During non-loop-closing operation (i.e. in areas where no loop closing is necessary or possible) only 50% of the processing power available is used. During loop-closings, the required processing power can jump to 500%. The exceeding 400% can now be delayed and distributed to be executed in one of the non-loop-closing phases. This way, all information can be used without the vehicle having to wait at loop-closing events.

Another idea for loop-closing was to use multi-threading similar to the way it was used for feature extraction (see section 4.2.4.3). This has the prospect of boosting the number of loop-closures which can be processed significantly, especially when the vehicle PC is upgraded to the latest 4-core 8-thread mobile processors.

Konolige (35) introduced the idea of skeleton frames. Faced with the problem of large graphs in graph-SLAM, his idea was to extract a skeleton graph from the complete graph which only contains the nodes necessary to keep the graph together

6. CONCLUSION

(hence skeleton frames). Loop closing is then only applied on the skeleton graph, and since its relation to the complete graph is known, can be back-propagated. The feasibility of this approach for the SURE-SLAM needs to be studied.

Long-Term Loop-Closing One of the questions regarding long-term feasibility of the algorithm was, if it could recognize areas it had previously visited if not only minutes/hours have passed but days or weeks. This “long-term loop-closing” should work as long as the environment did not change too severely. In order to put this to the test, the following experiment was planned for the Unisee environment: The vehicle was moving from a defined starting point at the pier to a survey location, doing a small-scale survey, and returning to the pier. Two weeks later the same experiment was to be conducted, and the data from the first experiment given to the algorithm as prior map. This way, ideally both a new map and its geometric relation to the old map should be computed during the mission. Due to problems with DAGON’s thrusters only the first half of the experiment could be conducted during this thesis, thus it has to be repeated in the future.

Active Navigation, Exploration Currently the SURE-SLAM algorithm is only used passively: it does not actively change the vehicle behavior, but only provides the navigation with localization and motion data. There are a number of ways to improve navigation by using further information from the algorithm: For example a warning can be sent to the navigation component if localization quality is deteriorating (e.g. due to increased turbidity) in order for the navigation to change direction to avoid loss of localization. Such behaviors can be summarized as “exploration feature”, and there is already work going into this direction (66), (54).

Long-Term Map Management As stated in section 4.2.4.4, currently the SURE-SLAM algorithm is not fit for long-term operation due to memory restrictions. There are a number of approaches to remedy this, all concerning detection of sub-maps, which then can be handled independently (i.e. stored on disk) and then joined (3). For graph-SLAM approaches this is relatively straightforward, especially if the trajectory to be executed is known beforehand.

6.5.3 Future of the AUV Dagon

The vehicle DAGON will continue to be used as experimentation platform for localization algorithms. Since it can now be seen as DFKI-RIC's best equipped vehicle with a completely functional localization suite in two modalities, it is the ideal platform for further development of algorithms in the underwater domain. In the near future, it will be used as main vehicle in the Europa-Explorer project mentioned above, until the exploration vehicle is finished and operational. After this, it will be used in the project "Trans-TerrA"¹, which aims to transfer space technologies to terrestrial applications. One of the application will include a docking system for underwater vehicles (based on works like (44)), and DAGON will be the test and carrier vehicle for this purpose. Preliminary experiments with a prototype were already conducted (64), laying the ground-work for extension and improvements.

¹<http://robotik.dfki-bremen.de/de/forschung/projekte/transterra.html>

6. CONCLUSION

References

- [1] Motilal Agrawal, Kurt Konolige, and M.R. Blas. Censure: Center surround extremas for realtime feature detection and matching. *Computer Vision/ECCV 2008*, pages 102–115, 2008.
- [2] Josep Aulinas, Marc Carreras, Xavier Llado, Joaquim Salvi, Rafael Garcia, Ricard Prados, and Yvan R. Petillot. Feature extraction for underwater visual SLAM. *OCEANS 2011 IEEE - Spain*, pages 1–7, June 2011.
- [3] Josep Aulinas, CS. Lee, Joaquim Salvi, and Yvan R. Petillot. Submapping SLAM based on acoustic data from a 6-DOF AUV. In *8th ifac conference on control applications in marine systems*, number 1988, 2010.
- [4] Josep Aulinas, Yvan R. Petillot, and X. Lladó. Vision-based underwater SLAM for the SPARUS AUV. In *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries (COMPIT'11)*, pages 171–180, 2011.
- [5] H. Bay, A. Ess, T. Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, June 2008.
- [6] James G. Bellingham, Thomas R. Consi, Usha Tedrow, and Diane Di Massa. *Hyperbolic Acoustic Navigation for Underwater Vehicles: Implementation and Demonstration*. Sea Grant College Program, Massachusetts Institute of Technology, 1993.
- [7] Paul J. Besl. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [8] N.A. Brokloff. Matrix algorithm for doppler sonar navigation. In *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.'* *Proceedings*, pages 378–383, 1994.
- [9] Hunter C. Brown, Ayoung Kim, and Ryan M. Eustice. An Overview of Autonomous Underwater Vehicle Research and Testbed at PeRL. *Marine Technology Society Journal*, 43(2):33–47, May 2009.
- [10] J. Choi, R.E. Curry, and G.H. Elkaim. Curvature-continuous trajectory generation with corridor constraint for autonomous ground vehicles. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 7166–7171. IEEE, 2010.
- [11] Leif Christensen, Martin Fritsche, Jan Albiez, and Frank Kirchner. USBL pose estimation using multiple responders. In *Oceans'10 Ieee Sydney*, pages 1–9. Ieee, May 2010.
- [12] Jason Cong, Muhuan Huang, and Yi Zou. 3D Recursive Gaussian IIR on GPUs and FPGAs. *cadlab.cs.ucla.edu*, pages 70–73, 2011.
- [13] Peter Corke, Carrick Detweiler, M. W. Dunnigan, Michael Hamilton, Daniela Rus, and Iuliu Vasilescu. Experiments with Underwater Robot Localization and Tracking. *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4556–4561, April 2007.
- [14] Nico Cornelis and Luc Van Gool. Fast scale invariant feature detection and matching on programmable graphics hardware. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. Ieee, June 2008.
- [15] M. Dunbabin, Peter Corke, and G. Buskey. Low-cost vision-based AUV guidance system for reef navigation. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, number April, pages 7–12. Ieee, 2004.
- [16] Ryan M. Eustice, Hanumant Singh, and John J. Leonard. Exactly Sparse Delayed-State Filters for View-Based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, December 2006.

REFERENCES

- [17] Ryan M. Eustice, Hanumant Singh, John J. Leonard, and M. R. Walter. Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters. *The International Journal of Robotics Research*, 25(12):1223–1242, December 2006.
- [18] Ryan M. Eustice, Hanumant Singh, and Oscar Pizarro. Visually Augmented Navigation for Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, 33(2):103–122, April 2005.
- [19] Nathaniel Fairfield, Dominic Jonak, George Kantor, and David Wettergreen. Field results of the control, navigation, and mapping systems of a hovering AUV. In *International Symposium on Unmanned Untethered Submersible Technology*, 2007.
- [20] Martin A. Fischler and Robert C. Bolles. Paradigm for Model. *Communications of the ACM*, 24(6), 1981.
- [21] S.D. Fleischer. *Bounded-error vision-based navigation of autonomous underwater vehicles*. PhD thesis, 2000.
- [22] Marcus Gary and Nathaniel Fairfield. 3D mapping and characterization of sistema Zacatón from DEPTHX (DEep Phreatic THERmal eXplorer). In *Proceedings of KARST08: 11th Sink-hole Conference ASCE*, pages 1–11, 2008.
- [23] N.R. Gracias and J. Santos-victor. Underwater Video Mosaics as Visual Navigation Maps. *Computer Vision and Image Understanding*, 79(1):66–91, July 2000.
- [24] C. J Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proc. Fourth Alvey Vision Conf.*, volume pp, pages 23.1–23.6. Alvey Vision Club, 1988.
- [25] Richard Hartley. Computation of the Quadri-focal Tensor. In *Lecture Notes In Computer Science*, volume 1406, pages 20–35. Springer-Verlag, London, 1998.
- [26] Richard Hartley and W. Zimmerman. *Multiple View Geometry in Computer Vision*, 2nd ed, volume 0521540518. 2004.
- [27] Marc Hildebrandt, Jan Albiez, Marius Wirtz, Philipp Kloss, Jens Hilljegerdes, and Frank Kirchner. Design of an Autonomous Under-Ice Exploration System. In *In MTS/IEEE Oceans 2013 San Diego, (OCEANS-2013)*, pages 1–6. IEEE, 2013.
- [28] Andrew Hogue, Andrew German, and Michael Jenkin. Underwater environment reconstruction using stereo and inertial data. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 2372–2377. Ieee, October 2007.
- [29] Viorela Ila, Juan Andrade-Cetto, Rafael Valencia, and Alberto Sanfeliu. Vision-based loop closing for delayed state robot mapping. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3892–3897. Ieee, October 2007.
- [30] Intel and WillowGarage. OpenCV Library, 2013.
- [31] M. Johnson-Roberson, Oscar Pizarro, Stefan Williams, and Ian Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.
- [32] Aaron J. Kapaldo. *Gyroscope Calibration and Dead Reckoning for an Autonomous Underwater Vehicle*. M.sc, University of Virginia, 2005.
- [33] Ayoung Kim and Ryan M. Eustice. Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency. *IEEE Transactions on Robotics*, 29(3):719–733, June 2013.
- [34] J.C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *{IFAC} Conference of Manoeuvring and Control of Marine Craft*, 2006.
- [35] Kurt Konolige and Motilal Agrawal. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, October 2008.
- [36] Rainer Kümmerle and G. Grisetti. g2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

- [37] Charles Loop and Zhengyou Zhang. Computing rectifying homographies for stereo vision. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, pages 125–131, 1999.
- [38] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol:60no2pp91–110, 2004.
- [39] Ian Mahon, Stefan Williams, Oscar Pizarro, and M. Johnson-Roberson. Efficient View-Based SLAM Using Visual Loop Closures. *IEEE Transactions on Robotics*, 24(5):1002–1014, October 2008.
- [40] J. Matas. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.
- [41] A. Milella and Roland Siegwart. Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *IEEE International Conference on Computer Vision Systems*, 2006.
- [42] L.G.B. Mirisola and J.M.M. Dias. Exploiting inertial sensing in visionbased mapping and navigation. Technical report, 2007.
- [43] F.A. Moreno, J.L. Blanco, and J. González. An efficient closed-form solution to probabilistic 6D visual odometry for a stereo camera. *Advanced Concepts for Intelligent . . .*, pages 932–942, 2007.
- [44] Aniket Murarka, Gregory Kuhlmann, Shilpa Gulati, Chris Flesher, Mohan Sridharan, and William C. Stone. Vision-based frozen surface egress: A docking algorithm for the ENDURANCE AUV. In *Proceedings UUST09, Conference on Un-manned, Un-tethered Submersible Technology*, number December, pages 22–25, 2009.
- [45] Shahriar Negahdaripour and Pezhman Firoozfam. An ROV stereovision system for ship-hull inspection. *Oceanic Engineering, IEEE . . .*, 31(3):551–564, 2006.
- [46] H. K. Nishihara. Practical Real-Time Imaging Stereo Matcher. *Optical Engineering*, 23(5), 1984.
- [47] Taragay Oskiper and Zhiwei Zhu. Visual odometry system using multiple stereo cameras and inertial measurement unit. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007.
- [48] J.P. Peyronnet. Posidonia 6000: a new long range highly accurate ultra short base line positioning system. In *OCEANS '98 Conference Proceedings*, pages 1721–1727, 1998.
- [49] Kristof Richmond. *Real-time visual mosaicking and navigation on the seafloor*. PhD thesis, 2009.
- [50] Rock. ROCK, the Robot Construction Kit, 2013.
- [51] Chris Roman, Oscar Pizarro, Ryan M. Eustice, and Hanumant Singh. A new autonomous underwater vehicle for imaging research. *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, 2000.
- [52] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. *Computer Vision/ECCV 2006*, pages 1–14, 2006.
- [53] Joaquim Salvi, Yvan R. Petillot, Stephen Thomas, and Josep Aulinas. Visual SLAM for underwater vehicles using video velocity log and natural landmarks. *Oceans 2008*, pages 1–6, 2008.
- [54] Robbie Shade and Paul M. Newman. Choosing where to go: Complete 3D exploration with stereo. In *2011 IEEE International Conference on Robotics and Automation*, pages 2806–2811. Ieee, May 2011.
- [55] Babak Ameri Shahrabi. *Automatic Recognition and 3D Reconstruction of Buildings through Computer Vision and Digital Photogrammetry*. Dissertation, University of Stuttgart, 2000.
- [56] Hanumant Singh, Ryan M. Eustice, Chris Roman, and Oscar Pizarro. The SeaBED AUVa platform for high resolution imaging. In *Un-manned Underwater Vehicle Showcase*, number Figure 1, 2002.
- [57] David A. Smallwood, Louis L. Whitcomb, and Senior Member. Model-Based Dynamic Positioning of Underwater Robotic Vehicles: Theory

REFERENCES

- and Experiment. *IEEE Journal of Oceanic Engineering*, 29(1):169–186, 2004.
- [58] Joe Stam. Stereo Imaging with CUDA. Technical Report January, NVIDIA, 2008.
- [59] J. Svab, T. Krajnik, J. Faigl, and L. Preucil. FPGA based Speeded Up Robust Features. In *Technologies for Practical Robot Applications, 2009. TePRA 2009. IEEE International Conference on*, pages 35–41. IEEE, 2009.
- [60] Stephen Thomas, Joaquim Salvi, and Yvan R. Petillot. *Real-time stereo visual SLAM for autonomous underwater vehicles*. Project thesis, Heriot-Watt University, 2008.
- [61] J. Tölgyessy. *Chemistry and Biology of Water, Air and Soil: Environmental Aspects*. Studies in Environmental Science. Elsevier Science, 1993.
- [62] Louis L. Whitcomb, Dana R. Yoerger, and Hanumant Singh. Combined Doppler/LBL Based Navigation of Underwater Vehicles. In *Proceedings of the the 11th International Symposium on Unmanned Untethered Submersible Technology*, number 9818. AUSA, 1999.
- [63] Stefan Williams and Ian Mahon. Simultaneous localisation and mapping on the great barrier reef. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1771–1776. IEEE, 2005.
- [64] Marius Wirtz, Marc Hildebrandt, and Christopher Gaudig. Design and test of a robust docking system for hovering AUVs. In *2012 Oceans*, pages 1–6. Ieee, October 2012.
- [65] Ruigang Yang and M. Pollefeys. A versatile stereo implementation on commodity graphics hardware. *Real-Time Imaging*, 11(1):7–18, February 2005.
- [66] Nan Ying and Low Eicher. Real-time 3D path planning for sensor-based underwater robotics vehicles in unknown environment. In *Oceans 2000 MTS/IEEE Conference and Exhibition*, number 65, 2000.
- [67] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 00, pages 0–7, 1999.

Declaration

I herewith declare that I have produced this thesis without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This thesis has not previously been presented in identical or similar form to any other German or foreign examination board.

The thesis work was conducted from 01.09.2009 to 20.12.2013 under the supervision of Prof. Dr. Frank Kirchner at DFKI Robotics Innovation Center Bremen.

Bremen, 20.12.2013