JACOBS
UNIVERSITY

An Intelligent and Robust System for Underwater Vision

by

Tomasz Łuczyński

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

Approved Dissertation Committee

Prof. Dr. Andreas Birk

Prof. Dr. Kaustubh Pathak

Dr. Nuno Gracias

Date of Defense: 8 December 2017

Computer Science & Electrical Engineering

Statutory Declaration

Family Name, Given/First Name	Łuczyński, Tomasz
Matriculation number	20331049
What kind of thesis are you submitting:	PhD-Thesis

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

This document was neither presented to any other examination board nor has it been published.

German: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde noch keiner anderen Prüfungsbehörde vorgelegt noch wurde sie bisher veröffentlicht.

……………………………………………………………………………………………………….

Date, Signature

Abstract

Cameras are one of the most common sensors in robotics. They are used in both research and industrial applications. Two cameras can be, and frequently are, combined into a stereo vision system in order to obtain additional 3D information. Underwater images suffer from extremely unfavourable conditions. Light is heavily attenuated and scattered. Attenuation creates change in hue, scattering causes blur and so-called veiling light. Furthermore, underwater images are distorted due to refraction through water-glass-air interfaces. All of this makes using cameras underwater exceptionally difficult.

Work presented in this thesis is mainly motivated by the needs of three EU-founded projects: MORPH, CADDY and DexROV. Work in these projects resulted in development at every stage in the design, calibration and image processing of underwater vision systems. This thesis is divided into five parts.

The first part introduces investigated research problems, and describes the background and motivation of the thesis. Furthermore, exemplar projects are described to give a good intuition of the state of the art and what advances have been made in this thesis.

The second part of this thesis describes the issue of refraction-based distortions. An extensive analysis of the problem is undertaken using a novel method, dubbed Pinax, that allows for very efficient and accurate modelling of submerged cameras. The Pinax model does not require any underwater calibration: a single in-air procedure is sufficient to handle a variety of underwater environments, including different salinities, temperatures and pressures. This method is shown to outperform any state of the art approach. At the same time, it remains much simpler and more practical.

The third part focuses on designing the stereo vision system from the perspective of selecting hardware and setup parameters that would perform best in a given task. This analysis is performed at a general level, so it may be applied for both in-air and underwater systems. Furthermore, 3D reconstruction error is discussed and a new way of modelling it is proposed. This knowledge was also incorporated into the hardware selection procedure.

Part IV addresses the problem of image degradation. The image formation process is discussed and an adaptation of the Dark Channel Prior to underwater conditions is proposed. This resulted with an image correction algorithm that allows for a reduction in backscattering in the registered images.

Part V describes of the practical applications of the methods presented earlier. The achieved results are discussed in terms of their real-life applications. Multiple examples are also provided. In addition to this, data generated by an underwater stereo system was further processed to generate a 3D grid map optimized for satellite communication, in order to demonstrate a full use case scenario related to the DexROV project.

Finally, Part VI summarizes the most important contributions and concludes this thesis.

Contents

List of Figures

List of Tables

Part I

INTRODUCTION

Historical Perspective

The beginnings of robotics may be traced back to early XX century and the Industrial Revolution. The idea of reorganizing production and replacing tedious human labour with machines was one of the greatest ideas of that time. However, the beginnings of modern robotics should rather be dated fifty years later, when first successful attempts of using electric servomechanisms in the design of manipulators were made. At the same time, transistor-based technologies were introduced to electronics. Shortly after, in 1961, the first Unimation robot was installed at a General Motors' casting plant. In 1962 multiple robots were acquired by Ford Motor Company. The year 1968 brought a commercial breakthrough in industrial robotics – that year as many as 48 robots were sold worldwide.

The development of mobile robots started with the cybernetic dog built by Henry Piraux in 1929. Significant progress in that field was made to meet the needs of the army during the Second World War. At that time the development focused on many variations of driving and flying bombs, e.g. V1 and V2 rockets had a basic autopilot and automatic detonation systems. Multiple research projects were conducted after the war leading to significant progress in mobile robotics and finding new possible applications. In 1969 a Mowbot was created. It was the very first robot that could automatically mow the lawn and it was aimed towards the average consumer.

Machine vision started to develop together with television. As early as in 1907 Boris Rosing, a Russian scientist and inventor, developed a photocell detector. The next major step in the field of computer vision was made in 1950, when the Vidicon tube was introduced – a device that could be considered the first digital camera. In 1960 Larry Roberts, often referred to as the 'father of computer vision', discussed the possibilities of extracting 3D geometrical information from 2D perspective views in his MIT PhD thesis. From this point on, machine vision started to expand as a separate field of research.

In its beginnings, underwater exploration was conducted utilizing manned submersibles. Such systems reached the peak of their popularity in the 1960s, when multiple US Navy contractors were developing systems of this kind. Safety concerns and technological progress led to the development of underwater robotics. The first tethered remotely operated vehicle (ROV), named POODLE, was developed in 1953 by Dimitri Rebikoff. The ROV concept evolved in the 1960s and the 1970s, and was developed mainly for military applications. In the 1980s ROVs gained popularity in numerous research applications and became essential for oil and gas industry when development of offshore technologies exceeded the reach of human divers.

It is difficult to say when different fields of research merged into what we presently know as robotics. It is no longer surprising to see a mobile platform carrying a manipulator or an industrial robot using computer vision to make decisions about its actions. Significant development in mobile robotics was possible thanks to technological advances in many fields, but two most important factors are the increase of computational power of modern computers and the development of new types of sensors. This allowed for fast and accurate acquisition of data as well as online processing of this data and reasoning. Typical mobile robots use various sensors to monitor their environment – from cameras to LIDAR sensors. A 3D model of a robot's surrounding is usually accurate and can easily be utilized to complete any given task.

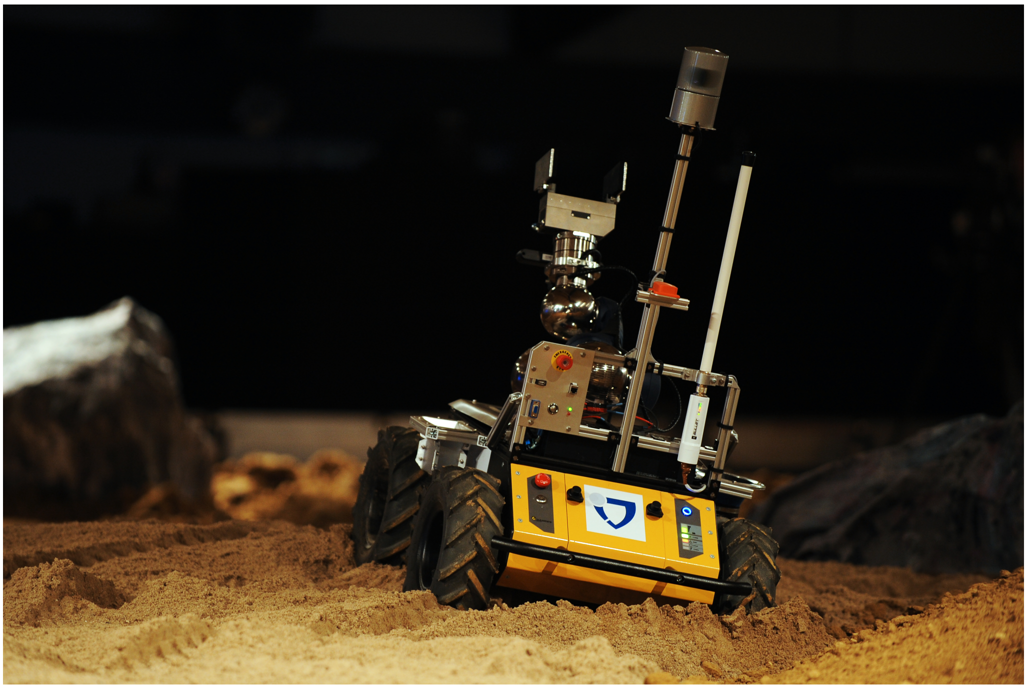


Figure 1: Mobile robot with manipulator. Robot from Jacobs University during SpaceBot Cup 2013.

Unmanned underwater vehicles (UUV) were initially developed separately from other mobile robots and also have to use different sensors. The conditions encountered under water make it impossible to port technologies developed for robots operating on the ground to UUVs. Electromagnetic waves are heavily attenuated by water, which makes it impossible to use GPS, WiFi or any other radio communication. Usually, all communications and the majority of sensing need have to be met using with acoustics. This results in significantly lower bandwidth and many problems not encountered when operating in normal atmosphere, e.g. the multipath effect and acoustic occlusions. Moreover, visible light is attenuated unequally, e.g. red light is absorbed much stronger than green light. As a result, underwater images, already recorded in low light conditions, have a tendency to shift their colours towards green-blue hue. Furthermore, all electronic equipment needs to be enclosed in sealed housings. In the case of cameras, this means that a glass panel in front of the lens may introduce refraction-based distortions. These need to be corrected if 3D reconstruction is to be performed. All of this makes underwater robotics very special: it utilizes a unique set of sensors and overcomes problems that are not present in normal atmosphere.

Chapter 1

Motivation and Research Goal

Over 70 percent of Earth is covered with water. The ocean floor is used for transporting oil and gas, mining, yet the underwater environment remains mostly unexplored. At present the most accurate map of the ocean floor ([Sandwell et al., 2014]) has a resolution of approximately 5km. This means that we know more about the surface of the Moon or Mars than about our oceans. Of course this does not apply to some small patches of seabed which were mapped precisely during research or industrial surveys. Furthermore, there is an urgent need for further development of underwater robotics as many tasks that are now performed by human divers are dangerous or cannot be performed with satisfying accuracy. Presently, this need for development is addressed by numerous research projects. Among these, there are three which are especially important from the standpoint of the following thesis, as they are the direct motivation for the work presented here: project MORPH, CADDY and DexROV.

1.1 Project MORPH

When surveying the marine environment the presently available techniques often fail to perform well enough or cannot be applied in their entirety. Traditionally, sensors are mounted on a ship or are towed by one. This way, a survey may be easily conducted over a large area. However, the costs of operating a ship with full crew and technical staff are very high. Furthermore, the available resolution is limited and some sensors, like cameras, may only be used over a short distance underwater. In some cases, when monitoring steep underwater cliffs, surveying from surface is impossible. Other operations, like monitoring the population of species are often performed manually, either by divers or using video footage. All of this could be greatly improved by using UUVs. The MORPH project ("Marine Robotic Systems of Self-Organizing, Logically Linked Physical Nodes", European Community's Seventh Framework Programme FP7 – under grant agreement n°. 288704) advanced the concept of surveying underwater environment using multiple separate vehicles cooperating for synergistic results. The robots form a physically disconnected, yet logically uniform structure. The formation may be morphed to match the terrain that needs to be investigated. Furthermore using different vehicles with different sensors allows for easy adaptation to the given task.

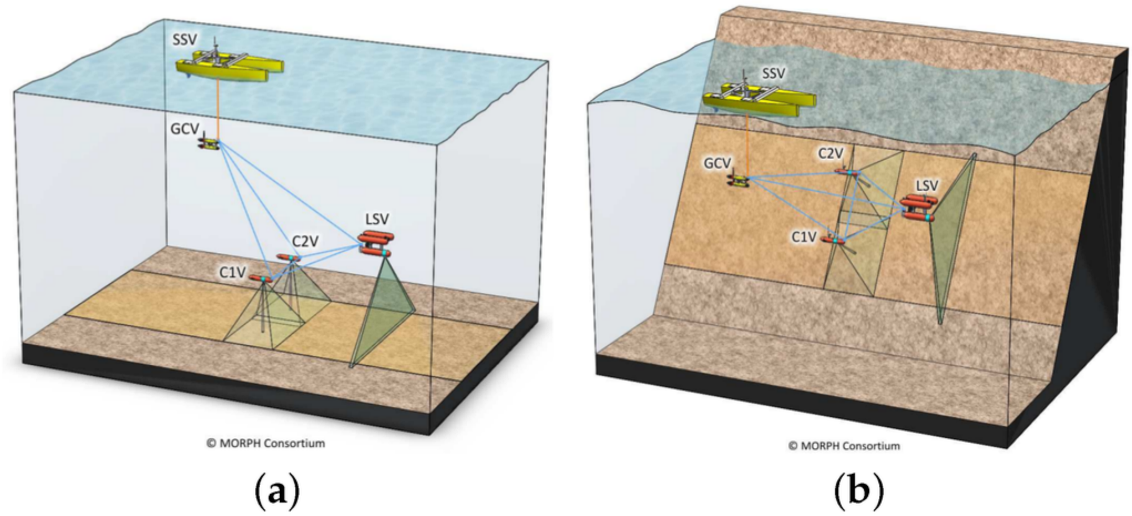


Figure 1.1: Example MORPH formations for mapping a flat sea bottom section and an underwater cliff. Acronyms describe robot's role in the formation: SSV – surface support vehicle, GCV – global communication vehicle, CV – camera vehicle and LSV – local sonar vehicle. [Kalwa et al., 2012]

1.2 Project CADDY

Project CADDY, "Cognitive Autonomous Diving Buddy", was a collaborative project funded under the European Community's Seventh Framework Programme FP7 – Challenge 2: Cognitive Systems and Robotics –grant agreement n° 611373. As stated in the official project's description: *"Divers operate in harsh and poorly monitored environments in which the slightest unexpected disturbance, technical malfunction, or lack of attention can have catastrophic consequences. They manoeuvre in complex 3D environments, carry cumbersome equipment, while performing their mission. To overcome these problems, CADDY aims to establish an innovative set-up between a diver and companion autonomous robots (underwater and surface) that exhibit cognitive behaviour through learning, interpreting, and adapting to the diver's behaviour, physical state, and actions."* The goal of the project was to replace a human diver with an autonomous underwater vehicle (AUV) which, together with an unmanned surface vehicle (USV), will assist and supervise any task performed by the diver. The CADDY system has been designed to operate in three basic modes ([Mišković et al., 2016]):

- "observer" buddy: continuously monitors the diver, his safety and needs;

- “slave” buddy: while performing various tasks divers may greatly benefit from a robot’s direct assistance. Bringing something from the surface, mapping the area, saving current coordinates or just taking a photo when needed are just some of the functionalities offered by the CADDY system;
- “guide” buddy: leads the diver through the underwater environment.

Figure 1.2: Three main functionalities of the CADDY system are ”observer”, ”slave” and “guide”.

1.3 Project DexROV

DexROV: ”Dexterous Undersea Inspection and Maintenance in Presence of Communication Latencies”, this European Community’s Horizon 2020 project was founded under grant agreement n° 635491 ([Gancet et al., 2015]). The goal of this project is to develop a set of hardware and software tools assisting ROV operations. ROVs are essential for many offshore operations, both commercial and scientific. Unfortunately, the expenditure associated with such operations is very high due to the costs of maintaining the ship and significant off-shore manpower. The DexROV project comprises a ROV system which is deployed from a vessel in the Mediterranean sea as well as an onshore Mission Control Centre (MCC) used for controlling and monitoring the ROV operations (see Fig. 1.3).

In the considered setup three different logical and spatial locations (nodes) should be considered: *onboard* the ROV, the operator *vessel* and *onshore* in the MCC. In all these nodes, specific hardware and data has to be present to enable their respective functionality, for example generating an environment representation with a 3D grid map:

- stereo image pairs are captured *onboard* the ROV and transmitted to the *vessel*
- a 3D grid map is generated on board the *vessel* using the stereo images and transmitted to the *onshore* MCC through a satellite link

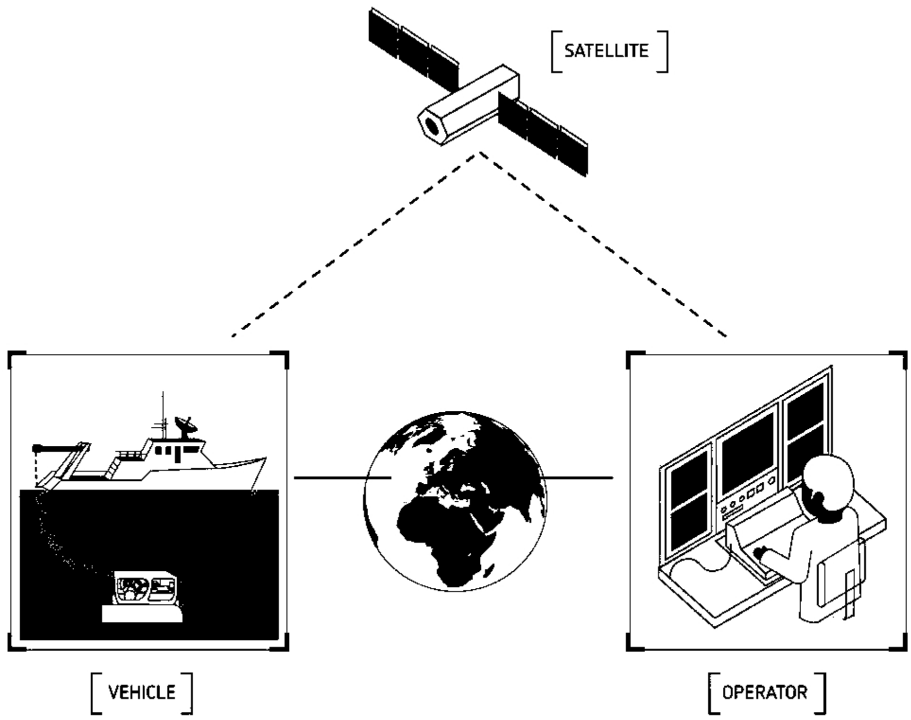


Figure 1.3: In the DexROV project the ROV will be operated via a satellite link from an onshore control centre.

- the *onshore* MCC displays the map for the pilots to orient themselves in the environment.

Other examples of data transmitted between the spatially distant nodes include the ROV's current location (*onboard* → *vessel*), the poses of objects autonomously recognized by machine perception (*vessel* → *onshore*) or motion commands (*onshore* → *onboard*).

This way, costs of ROV operations may be significantly reduced by moving some of the highly trained crew members to and onshore location. Furthermore, the skills of these people may be used more efficiently, e.g. they may work with one ROV when another one is on its way to the deployment-site. On the other hand, an ROV needs to be equipped with an additional perception system, to see, understand and support the operator. This is especially important when it comes to tasks like manipulation, where the operator cannot react quickly enough due to communication latency.

1.4 Research Questions

The three projects mentioned earlier require a robust vision systems allowing for precise 3D reconstruction of underwater environment. In a normal atmosphere, these specific requirements would be met by almost any stereo vision system. However, underwater conditions bring about numerous problems that need to be addressed.

Therefore, the main subject of this thesis is the *full process of designing an underwater stereo vision system* that will match the capabilities of a system operating in a normal atmosphere, overcoming all the difficulties and meeting the requirements of any given project. Specifically the following questions are addressed:

1. *How to design an underwater stereo system?* This question relates to optimum selection of hardware and setup parameters that will guarantee the required accuracy and density of the 3D reconstruction.

2. *How to improve the quality of underwater images?* "Improvement" should be understood in context of removing haze from the image for best 3D representation.

3. *How to handle refraction-based distortions introduced by glass panels in the camera housing?* A solution that is easy to implement in real life conditions and allows for closed-form 3D reconstruction is especially desirable.

1.5 List of Publications

The results described in this thesis have been published or contributed to the following papers:

1. Łuczyński, T., Pfingsthorn, M., and Birk, A. (2017). The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings. *Ocean Engineering*, 133:9 – 22.

2. Łuczyński, T., Pfingsthorn, M., and Birk, A. (2017). Image rectification with the pinax camera model in underwater stereo systems with verged cameras. In *OCEANS 2017 - Anchorage*.

3. Łuczyński, T. and Birk, A. (2017). Underwater image haze removal with an underwater-ready dark channel prior. In *OCEANS 2017 - Anchorage*.

4. Łuczyński, T., Fromm, T., Govindaraj, S., Mueller, C., and Birk, A. (2017). 3d grid map transmission for underwater mapping and visualization under bandwidth constraints. In *OCEANS 2017 - Anchorage*.

5. Pfingsthorn, M., Rathnam, R., Łuczyński, T., and Birk, A. (2016). Full 3d navigation correction using low frequency visual tracking with a stereo camera. In *OCEANS 2016 - Shanghai*, pages 1–6.

6. Enchev, I., Pfingsthorn, M., Łuczyński, T., Sokolovski, I., Birk, A., and Tietjen, D. (2015). Underwater place recognition in noisy stereo data using fab-map with a multimodal vocabulary from 2d texture and 3d surface descriptors. In *OCEANS 2015 - Genoa*.

7. M. Benndorf, M. Garsch, C. Mueller, T. Fromm, T. Haenselmann, N. Gebbeken, T. Łuczyński, and A. Birk. Robotic bridge statics assessment within strategic flood evacuation planning using low-cost sensors. *SSRR Conference - Shanghai*, 2017.

1.6 Structure of the Thesis

The order in which results are described in this thesis does not follow the order of research questions presented in Sec. 1.4. A different organization of topics is necessary as some findings listed in one part are a basis for discussion in the others. The remaining part of this thesis is organized in the following way. Part II analyses the influence of refraction on the camera model and image distortions. Then, a new camera model is proposed, discussed and tests are described. Part III investigates the process of designing a stereo system, taking into account the desired accuracy, 3D reconstruction capabilities and various hardware options. Part IV focuses on underwater image creation model and offers a method for haze removal, as this factor disturbs 3D reconstruction the most. Part V describes major use cases of algorithms described in this thesis. In some chapters this description is extended with additional contributions specific to those applications. Finally Part VI summarizes the most important contributions and concludes this thesis.

Chapter 2

Underwater Machine Vision

2.1 Generic Camera Models in Machine Vision

In Part II, a novel way of modelling the camera underwater is presented. This model combines the projection function of the physically accurate, axial model together with the desirable pinhole camera model properties. To put this work into perspective, the generic camera models will be shortly summarised here. As full descriptions of all camera models would be far too long and are not relevant for the rest of this thesis, only a summary is presented in this section. Detailed information about each camera model may be found e.g. in [Sturm et al., 2006].

In machine vision the most popular way of modelling the camera is with the pinhole camera model (also known as central or perspective camera). This model assumes that all light rays forming the image cross one point called the centre of projection. Even though the path taken by light as it travels through the lens may be much more complex, the registered image can be described with this simple model. Some additional deformations in the image, which occur due to the use of a lens instead of a physical pinhole (for example in camera obscura), may be corrected with the lens distortion model. This approach guarantees easy mathematical description of the projection of 3D points to the image plane, and is also an easy way of finding the light rays in the 3D plane, corresponding to a given pixel.

The second generic camera model, called the axial model, assumes that all the light rays forming an image do not necessarily cross a single point, but instead all cross a line ([Ramalingam et al., 2006]). This model may be applied to different cameras, e.g. a stereo camera setup may be treated as an axial camera - compare Fig. 2.1.

The axial camera model is especially important for this thesis as the submerged perspective camera using flat glass panel housing was in fact identified as an axial camera [Agrawal et al., 2012]. This needs to be taken into account during the calibration process, and additionally when using underwater images for the 3D reconstruction of the observed scene later on.

Even though these two camera models cover a great majority of cameras being

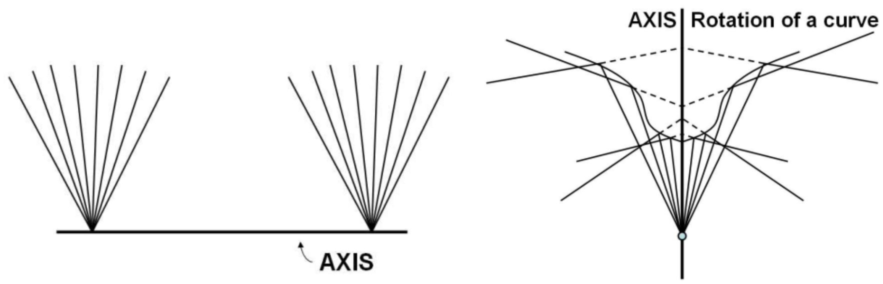


Figure 2.1: Examples of axial imaging models. Left: stereo camera setup may be treated as an axial camera, as all the light rays forming the images cross one line. Right: a mirror formed by rotating a planar curve about an axis containing the centre of projection of the perspective camera. Image reprinted from [Ramalingam et al., 2006].

Table 2.1: Generic camera models and their basic characteristics

Points/lines cutting the rays	**Description**
None	Non-central camera
1 point	Central camera
2 points	Camera with a single projection ray
1 line	Axial camera
1 point, 1 line	Central 1D camera
2 skew lines	X-slit camera
2 coplanar lines	Union of non-central 1D camera and central camera
3 coplanar lines without a common point	Non-central 1D camera

applied in machine vision, other models exist. The summary of these models is presented in the Table 2.1 (reprinted from [Sturm et al., 2006]).

This classification is very generic and covers all possible vision systems. A different categorisation may be used to emphasise other aspects, e.g. construction of the vision system. Catadioptric systems are a good example: these cameras consist of a pinhole camera and a mirror in which the scene is reflected. Some additional lenses in front of the mirrors may also be added. These cameras offer very wide field of view ([Nayar, 1997]) or large magnification in a compact form ([Bahrami and Goncharov, 2010]). Within the classification presented here, catadioptric cameras may be either non-central or axial cameras, depending on the shape and the placement of the mirror.

2.2 Challenges in Underwater Machine Vision

When working with underwater machine vision there are unique problems that need to be addressed. Due to these challenges, reusing methods and algorithms developed for images registered in air is often impractical or even impossible.

Most of the challenges that must be overcome arise from the physical properties of water and cannot be omitted in any way. Water strongly absorbs electromagnetic waves, making wireless communication via Bluetooth or WiFi impossible. The visible spectrum is also affected; different wavelengths of light are absorbed at different rates, leading to shifts in colour, usually towards the blue end of the spectrum. The overall intensity of light is also reduced. Additionally, light in water is being scattered, causing blur and haze in the images. This leads not only to inaccurate colour representation but low quality of the images in general. This issue is discussed in detail in Part IV of this thesis. A possible way of dealing with hazy images is also proposed.

A second group of challenges arises from the fact that the cameras must be physically protected from water and are enclosed in housings. Different constructions of housing may be used, but there are two main choices for the glass through which the camera will observe the environment: it is typically flat or spherical (compare Fig. 2.2). Both options have their advantages and disadvantages. A spherical port will not introduce any additional distortions to the image if the camera is positioned so that the centre of projection is in the centre of the sphere. This, however, is not easy to guarantee and requires tedious manual calibration. Furthermore, each time the housing is opened, e.g. for maintenance, the calibration process must be repeated. The glass spheres used in such ports are also not easy to manufacture and are therefore expensive. Finally, cameras are usually placed in front of the ROV / AUV and spherical glass is therefore exposed to physical damage in this position.

Figure 2.2: Two typical constructions of the camera housing: with flat glass panel (left) and with spherical glass (right).

A flat glass panel on the other hand is easy and cheap to manufacture, can be easily used in many housing designs and is not as exposed as a spherical port. Unfortunately the image registered through a flat glass panel will suffer from significant image distortions introduced by refraction through water-glass and glass-air interfaces. Moreover, it was shown that the pinhole camera model is not valid in this setup ([Treibitz et al., 2012]). This problem is addressed in Part II of the thesis, and a new way of modelling a submerged camera behind a flat glass panel is proposed, allowing for easy and accurate image rectification.

The final group of challenges is caused by the conditions in which the data is collected both for testing and in regular use. Limited communication with the vehicle, limited power and overall harsh conditions make data collection much more difficult than in air. Usually it is not feasible to record enough data to use deep learning techniques. Furthermore, some camera interfaces, e.g. Fire Wire, have limited cable length (4.5m in the case of Fire Wire). As high bandwidth wireless communication is not possible underwater, recorded images need to be stored or processed on board of the vehicle. This adds yet another electrical component that needs to be in housing, protected from the water. Some procedures, such as camera calibration, are also very challenging underwater, as the diver performing the calibration will generally not have any visual feedback. It is very hard to get any reasonable calibration data this way.

In conclusion, there are many unique challenges that need to be addressed in underwater machine vision. A system designed for underwater application is expected to work robustly against low light, high noise and overall unfavourable conditions. Many of these problems need to be addressed online, on board of the vehicle, hence some machine intelligence is expected to process the images reliably. Subsequent parts of this thesis deal with many of the challenges discussed here, proposing possible solutions.

Part II

REFRACTION CORRECTION

Chapter 3

Overview

3.1 Summary of the Research

Cameras are commonly used in underwater applications. This includes ship hull, pipeline and other inspection missions [Hollinger et al., 2012,Kim and Eustice, 2013, Foresti, 2001, Asakawa et al., 2000, Negahdaripour and Firoozfam, 2006, McLeod et al., 2013, Galceran et al., 2014], habitat mapping [Davie et al., 2008, Bodenmann et al., 2013], vehicle station-keeping [Negahdaripour and Fox, 1991, Marks et al., 1994,Lots et al., 2000], archaeology [Bingham et al., 2010,Chapman et al., 2010,Hue et al., 2011] or search and recovery missions [Purcell et al., 2011] just to name a few examples. A short overview with respect to underwater vision on unmanned underwater vehicles (UUV) is given, e.g. in [Horgan and Toal, 2006]. Flat-panel glass windows are often used for underwater camera housings. While domes provide optical advantages, they have to be specially engineered to fit the camera and the integration is not trivial. Flat-pane windows are hence simply a much less expensive and more flexible choice. On the other hand, flat ports introduce significant distortions due to the refraction at the air-glass and glass-water interfaces.

Most of the results presented in this part of the thesis have been published before in Ocean Engineering Journal, in the article: *"The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings"*([Łuczyński et al., 2017c]). Later publication in the proceedings of the IEEE Oceans'17 conference, titled *"Image rectification with the pinax camera model in underwater stereo systems with verged cameras"* ([Łuczyński et al., 2017b]), complemented these findings with analysis of the case when verged cameras are placed behind a single glass panel and is also presented here. In addition to the previously published results some extended analysis and simulations for different cases are provided in this part of the thesis.

Two main contributions are made. Firstly, the problem of underwater camera modelling from a practitioner's viewpoint is discussed. Illustrative examples of the underlying effects and their relevance to real world applications are provided. To some extent, this also bridges apparent contradictions found in the literature that can be explained when contrasting theoretical considerations with typical application cases. Secondly, a novel approach for calibration and refraction correction of underwater images is provided. Proposed method is very convenient to use in real world applications being very accurate at the same time. This model, dubbed *Pinax*, is based on a virtual *pinhole* camera model - which is demonstrated herein to be applicable for real world underwater housings where the camera is relatively close to the flat-pane - while using the projection function of an *axial* camera. The Pinax model incorporates the water refraction index, for which - as also experiments show - it is sufficient to derive it through (estimated) salinity to achieve accurate results. It is hence sufficient to calibrate the underwater camera only once in air, thus replacing tedious in-water calibrations before or during missions. For the rectification, a look-up table is generated using the projection function of the axial model, for which we show that it can be used in a significantly simplified fashion within the Pinax model. The look-up table can be easily computed a priori and allows very fast real-time refraction correction of single images. An alternative method of representing correction maps, via lens distortion coefficients, is also discussed. Real world experiments with different cameras in different fresh and salt water environments show that the Pinax model outperforms standard methods.

3.2 State of the Art Discussion

The predominant way to handle the refraction-based distortions is to use a standard perspective projection model and to perform standard camera calibration in-situ, i.e., in the water or by including estimated correction factors, see e.g., [Shortis and Harvey, 1998, Gracias and Santos-Victor, 2000, Pessel et al., 2003, Pizarro et al., 2003, Lavest et al., 2003, Negahdaripour et al., 2006, Negahdaripour et al., 2007, Brandou et al., 2007, Sedlazeck et al., 2009, Johnson-Roberson et al., 2010, Kunz and Singh, 2010, Beall et al., 2011, Kang et al., 2012].

[Treibitz et al., 2008, Treibitz et al., 2012] show that flat port cameras do not possess a single viewpoint (SVP), i.e., the perspective projection model is invalid for flat ports. This is also supported by other works [Li et al., 1997, Kunz and Singh, 2008, Chari and Sturm, 2009, Gedge et al., 2011, Yamashita et al., 2011, Sedlazeck and Koch, 2011, Jordt-Sedlazeck and Koch, 2012, Agrawal et al., 2012, Servos et al., 2013, Jordt-Sedlazeck and Koch, 2013, Chen and Yang, 2014, Jordt-Sedlazeck and Koch, 2012, Yau et al., 2013].

In [Kunz and Singh, 2008] the errors caused by not compensating the refractive distortions are discussed in some detail and they are identified as significant, however no solution to this problem is presented. A mathematical model of underwater imaging through planar glass ports is introduced in [Chari and Sturm, 2009]. Matrices corresponding to fundamental and homography matrices are derived. They, however, depend on the incident angle of the light ray corresponding to each image pixel, so they cannot be used directly for underwater vision methods. Since no continuation of this work was published, their results remain as theoretical considerations of conceptual value. Apart from analysing the general problem from a largely theoretical perspective, [Treibitz et al., 2008, Treibitz et al., 2012] provide an approach for a single refractive layer, i.e., when the window is negligibly thin and the problem can be reduced to only a single air-water interface.

Important insights into the problem and ways towards a solution are presented in [Agrawal et al., 2012] where a flat port camera is identified to be in fact an axial camera. [Agrawal et al., 2012] derive a 12th degree polynomial that must be solved to project a 3D point onto an image plane in this case. A method is proposed for calibration of the camera but it requires knowledge of the full 3D geometry of the calibration points in the environment - a requirement which is difficult if not impossible to fulfil in underwater applications. Furthermore, the underlying axial model does not allow for a rectification of single images as the axial model implies that the points are lying on complex curves. Correspondences across multiple images can, in principle, be exploited, but this is computationally very complex, as also pointed out in [Jordt-Sedlazeck and Koch, 2013].

When using multiview methods, the SVP model can lead to reasonable results as explicitly discussed in [Kang et al., 2012]. Nevertheless, [Jordt-Sedlazeck and Koch, 2013] rely on the results from [Agrawal et al., 2012] by proposing a refractive Structure from Motion (SfM) method by augmenting the standard perspective SfM process by incorporating a new error function in the optimization and report clear improvements. In their further work [Jordt et al., 2016] integrated [Jordt-Sedlazeck and Koch, 2013] and [Jordt-Sedlazeck et al., 2013] into a complete system for refractive reconstruction by improving the non-linear optimization. This allowed them to use their method on larger scenes. While this is an interesting approach, it requires sufficiently many images with sufficiently different views of the scene and it is still computationally very demanding.

Chapter 4

Problem Formulation and Preliminary Analysis

4.1 Flat-Panel Camera Setup

The following setup is considered. A physical camera ${}^{p}Cam$ that follows the standard SVP (single viewpoint) model with an intrinsics matrix $\mathbf{{}^{p}K}$ is enclosed in a water sealed housing with a flat glass panel through which it observes the underwater environment. The glass panel is flat and both sides are parallel. The glass panel introduces distortions that are to be handled by a virtual camera model ${}^{v}Cam$ that interprets the environment scene from the physical camera ${}^{p}Cam$. The overall underwater setup of the physical camera plus its housing with a flat-pane window submerged into water is denoted as the underwater camera ${}^{u}Cam$. When the underwater camera is in air, e.g., for the calibration, it is denoted with ${}^{u}_{a}Cam$.

If not mentioned otherwise, in this part of the thesis the term *camera* refers to the complete underwater set-up and terms *virtual camera* and *physical camera* refer to the model ${}^{v}Cam$ of the glass-panel refraction, or to the in-air physical device ${}^{p}Cam$ inside the housing, respectively.

The main object of interest for this part of the thesis is defining the virtual camera model ${}^{v}Cam$ to handle the refraction induced distortions. The related notations and a schematic view are presented in Fig. 4.1. Following parameters are used:

- d_0 - distance from the centre of projection of ${}^{p}Cam$ to the glass window,
- d_1 - thickness of the glass,
- x - distance to point of intersection of the light ray with the camera axis,
- Δx - length of the focus section (discussed in more detail in Sec. 4.3),
- n_a, n_g, n_w - refraction indexes (scaled so that $n_a = 1$),
- $\mathbf{n}$ - normal vector to the glass surface,

- α - incident angle.

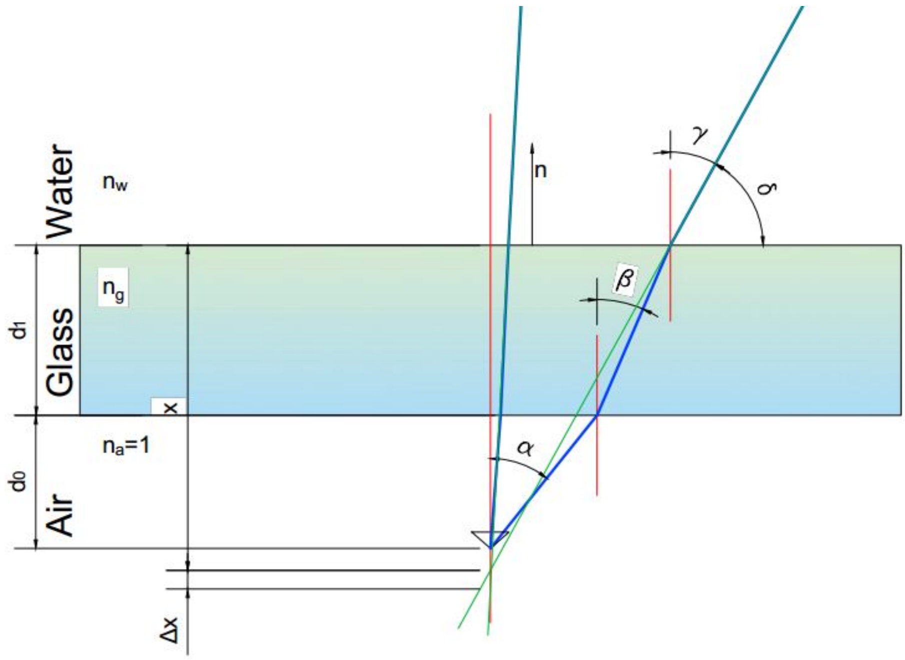


Figure 4.1: Schematic view of a Flat Port setup: d_0 - distance from the centre of projection to the glass window, d_1 - thickness of the glass, x - distance to point of intersection of the light ray with the camera axis, Δx - length of the focus section, n_a, n_g, n_w - refraction indexes, scaled so that $n_a = 1$, $\mathbf{n}$ - normal vector to the glass surface, α - incident angle. The blue line represents the physically accurate light ray; the green line is the apparent ray traced back to the camera's optical axis.

4.2 The Flat Port Setup as an Axial Camera

As shown in [Agrawal et al., 2012], the physically accurate model of a flat-port underwater camera corresponds to an axial camera model. So, light rays creating the image do not intersect in one point, as in the SVP pinhole model, but they all intersect one line, called the axis of the camera. Using the pinhole camera model thus requires approximating the focus section, i.e., the line segment on the axis on which rays cross, with a single point. The conclusion is that the quality of this approximation depends directly on the length Δx of this section. In the limit case, the pinhole camera can be seen as an axial camera where the focus section of the camera axis is infinitesimally short. To analyse the refraction, ray tracing through the air-glass-water interface and the apparent intersection of the rays in the water with the camera axis can be modelled (compare Fig. 4.1):

$$\beta = \arcsin \frac{\sin \alpha}{n_g}$$

$$\gamma = \arcsin \frac{\sin \alpha}{n_w}$$
$$\delta = \frac{\pi}{2} - \gamma$$

For the sake of simplicity, it is assumed that the refractive plane normal and therefore the camera axis in the axial model is parallel to the optical axis of the camera. This assumption is without loss of generality since the incident angle α, i.e., the only parameter related to camera rotation, is one of the inputs, which can be easily rotated by a fixed off-set. For the sake of completeness, the equations for finding incident angles α, given the camera pose in the housing is:

$$\mathbf{v_0} = \mathbf{K}^{-1}\mathbf{p}$$
$$\alpha = \arccos \frac{\mathbf{v_0}^T\mathbf{n}}{|\mathbf{v_0}||\mathbf{n}|}$$

where $\mathbf{K}$ is the intrinsic parameter matrix and $\mathbf{p}$ represents pixel coordinates on the image.

The focus distance x for each light ray (Fig. 4.1) can be computed as:

$$x = \tan \delta (d_0 \tan \alpha + d_1 \tan \beta)$$

4.3 Length of the Focus Section

Consider an example setup with a glass refraction index $n_g = 1.5$, a water refraction index of $n_w = 1.335$ and a glass thickness of $d_1 = 10mm$. Plotting the change of x as a function of the incident angle α and of the distance d_0 illustrates a very important aspect (Fig. 4.2). As d_0 grows, the changes in the focus distance depending on the incident angle α (along X axis) become more significant, i.e., there is a higher range of focus distances with increasing d_0. This is further illustrated in the following.

Fig. 4.3 shows the length of the focus section Δx as a function of d_0 and d_1. Δx was calculated by finding distance x for different incident angles α and finding the maximum distance between registered values. The plot shows that changes in d_0 are much more significant than changes in d_1, i.e., the distance of the camera to the flat-pane window has a stronger effect than the thickness of the glass window. This effect is caused by a relatively small difference between the refraction index of glass (≈ 1.5) and the average water refraction index (≈ 1.33, [Roswell et al., 1976]) compared to the more significant refraction on the glass-air interface. In Fig. 4.3 it can be also observed that the best approximation of the axial camera model with a pinhole model occurs for small values of d_0.

To further motivate and illustrate this, Fig. 4.4 shows where the light rays in water cross the camera's optical axis for different values of d_0. Each line on the graph corresponds to a different incident angle ranging from 0 to 35 degrees, i.e., a physical camera with a field of view of 70 degrees. It can be seen that they never cross the same spot, but for some optimal d_0, they are very close to intersecting in one point. Points of intersection of the light rays in water and the camera axis move differently, depending on the incident angle α that they correspond to. They get

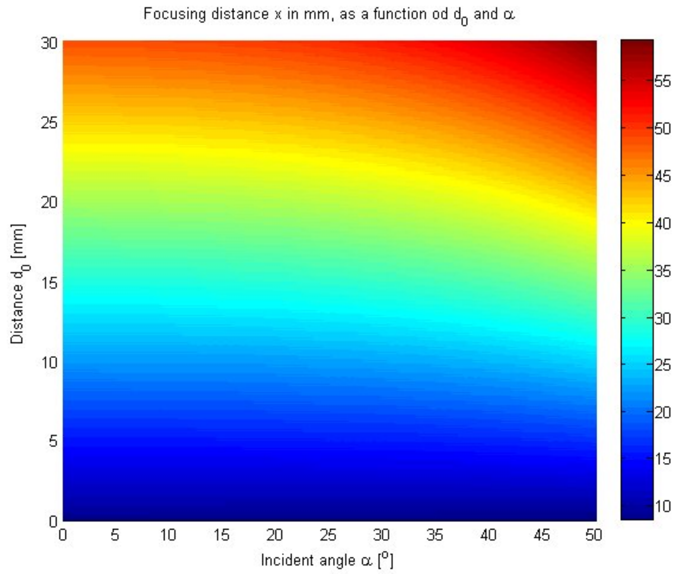


Figure 4.2: The focus distance x (in mm) as a function of d_0 and α for an example setup with a glass refraction index of $n_g = 1.5$, a water refraction index of $n_w = 1.335$ and a glass thickness of $d_1 = 10mm$. It can be seen that the changes in the focus distance x for different incident angles α become more significant for increasing distances d_0 of the physical camera to the flat-pane.

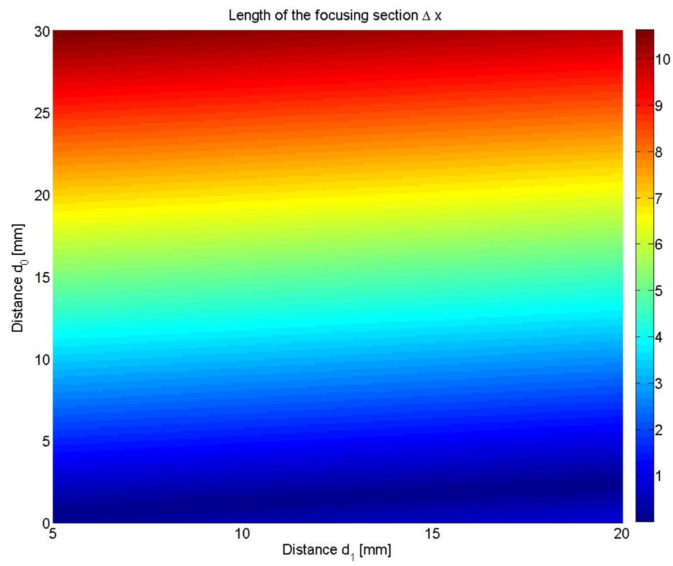


Figure 4.3: The length of the focus section (Δx) as a function of d_0 and d_1. It can be seen that the influence of the distance d_0 of the physical camera to the flat-pane is more significant than the thickness d_1 of the glass-pane.

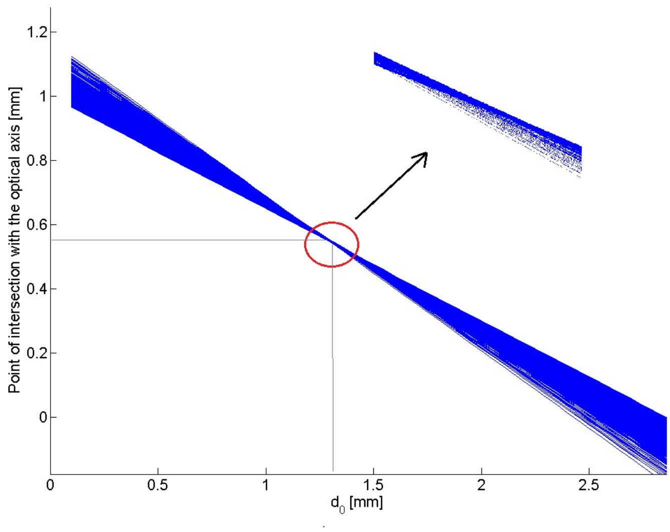


Figure 4.4: Example distances where the light rays traced back from the water cross the optical axis of the camera depending on d_0. Different lines correspond to different incident angles ranging from 0 to 35 degrees, i.e., a physical camera with a field of view of 70 degrees.

closer to each other and then they move past each other increasing the value of the Δx again. To find this optimal value of d_0 for some given parameters the following method is used. We implemented ray tracing based on the above formulation of the model. Then non-linear optimization is used to minimize the length of the section where light rays back-traced from the water intersect with the camera optical axis. For example, for the case where $d_1 = 10mm$, the glass refraction index $n_g = 1.5$, and the water refraction index $n_w = 1.335$, the method converges to $d_0 = 1.4282mm$ where all light rays intersect the optical axis on a section Δx that is only $0.0079mm$ long, i.e., within a very good approximation of a single point. The result of this numerical analysis allows us to define the middle of this section Δx as a secondary centre of projection placed $0.5851mm$ away from the glass panel. For this case, the virtual camera can be treated as an SVP camera and represented with the pinhole model.

This example motivates that although the pinhole camera model does not represent the actual physical state, for the purpose of underwater vision it may be used as basis for a model if the distance between the centre of projection of ${}^{p}Cam$ and the glass plane is very small. This is a realistic assumption as there are no reasons to design excessive housing sizes, i.e., the physical camera inside a housing is usually placed quite close to the flat-pane window.

4.4 Influence of the Distance of the Camera to the Flat-Panel

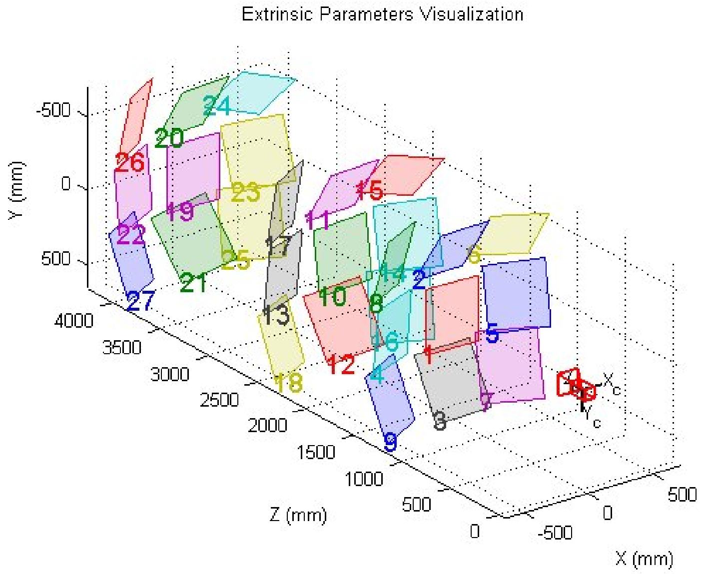


Figure 4.5: Poses of the simulated calibration patterns used for analysing the influence of the d_0 on the pinhole model accuracy.

The influence of the distance of the physical camera to the flat pane is now further illustrated in a other motivational example. The thickness of the glass panel d_1 is assumed to be constant at $d_1 = 10mm$. For different values of d_0, the camera is calibrated with a standard procedure. The calibration input data is based on 27 simulated chequerboard in 3D space (Fig. 4.5) by projecting the corner points to the image plane using the full physical model including refraction. This forward projection requires solving the twelve-degree polynomial introduced in [Agrawal et al., 2012]. This data is hence used to calibrate the camera as if it were underwater.

In a second step, a set of 100 random 3D points in front of the underwater camera is generated. This set is then projected onto the image plane twice for each d_0. Firstly, the projection is performed with the full physical model to get the expected image coordinates and the second time with the camera matrix from calibration using the pinhole model including undistortion. Then the distance between corresponding points (also called reprojection error) is calculated and used to evaluate the pinhole approximation for the different d_0. Fig. 4.6 and 4.7 show the results by plotting the average distance between corresponding points for the different d_0 values.

It should be noted that here values of d_0 up to $500mm$ are considered. Values within this range can be found in the literature for experimental setups, especially in the highly relevant works of [Agrawal et al., 2012] and [Treibitz et al., 2012]. Such big values are used because then the effects of the axial camera model are

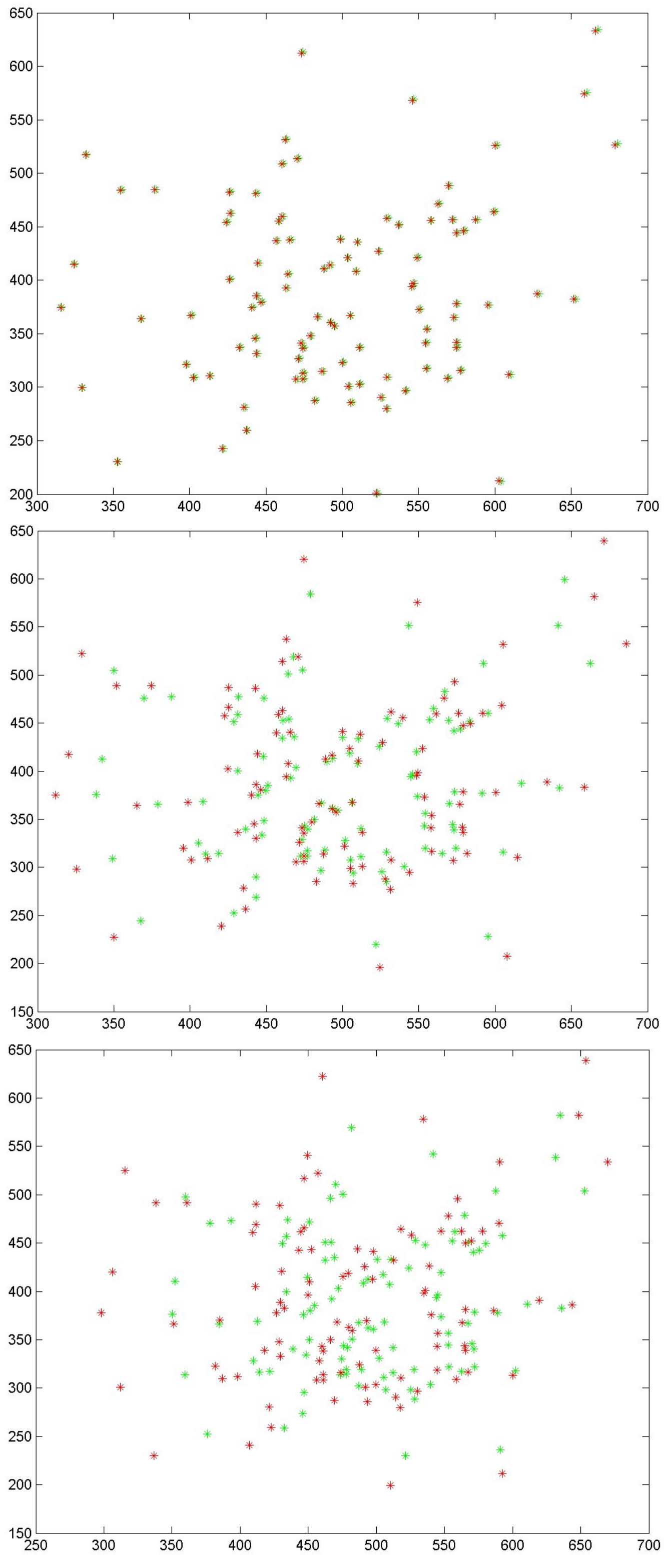


Figure 4.6: An example with a refraction-based (green) and a pinhole (red) projection of random points in the scene for $d_0 = 1mm$ (top), $300mm$ (centre), and $500mm$ (bottom) respectively. Please note the increasing deviations in the models with increasing d_0.

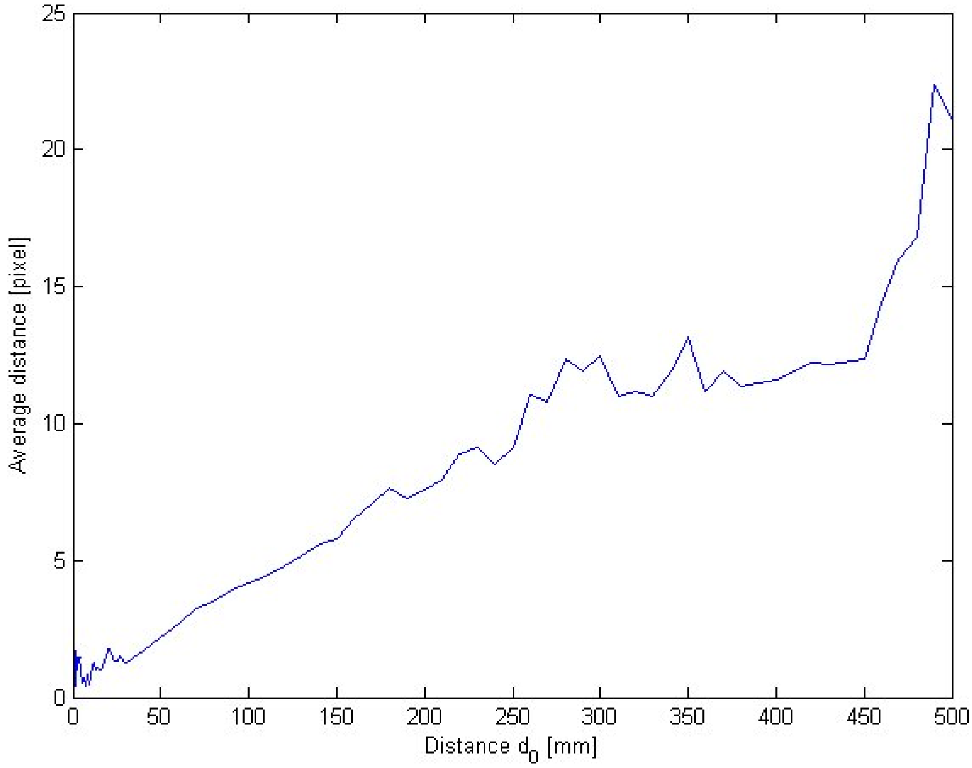


Figure 4.7: An example of the reprojection error for changing d_0. Note that for $d_0 = 0 - 10mm$ the errors caused by the SVP approximation can be neglected. The graph is not as smooth as may be expected, e.g., as in Fig. 4.3, because the simulated patterns were not always in the optimal positions for calibration, e.g., they did not always cover the whole field of view of the camera - which is a very natural effect that can also be observed in real world conditions.

clearly visible and for example the position of the camera in the housing can be found with nonlinear optimization. There may be applications where the distance of the physical camera to the flat-pane is quite large, e.g., when observing objects in an aquarium and the physical camera needs to keep a significant clearance to the aquarium window for some reason. However, this scenario is very unrealistic for underwater cameras. Excessive housing sizes to allow for significant distances d_0 are neither necessary nor desirable for underwater applications.

4.5 Influence of the Thickness of the Panel

Second experiment was performed in the same way as in Sec. 4.4, except this time d_0 was fixed at $1mm$ and the tested parameter was d_1. Results are presented in Fig. 4.8.

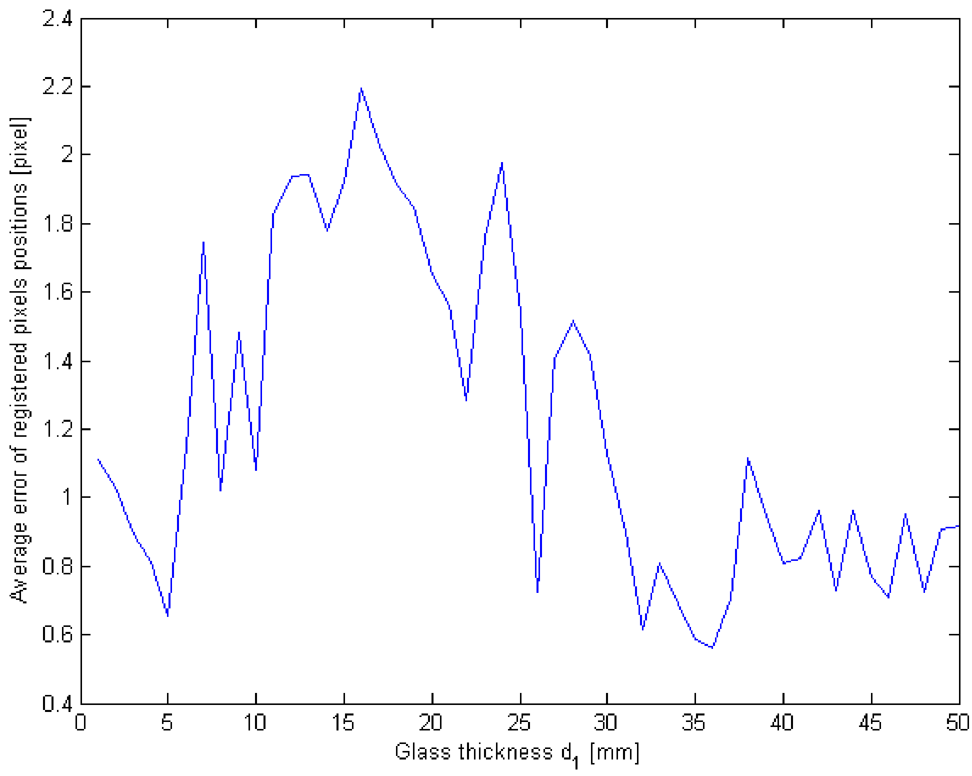


Figure 4.8: Reprojection error for changing d_1. Local maxima are caused by differences in the calibration quality for each tested setup and should not be interpreted as any meaningful result.

This second experiment also shows expected results. The thickness of the glass panel d_1 has much smaller influence on the reprojection error in comparison to d_0. Taking into account that this value will stay in the range of $1 - 20mm$ for most applications, it is safe to ignore this parameter as a source of error. Rapid changes of reprojection error on Fig. 4.8 are caused by local optima in the simulated camera calibration.

4.6 Rectification Accuracy near the Calibration Distance for SVP

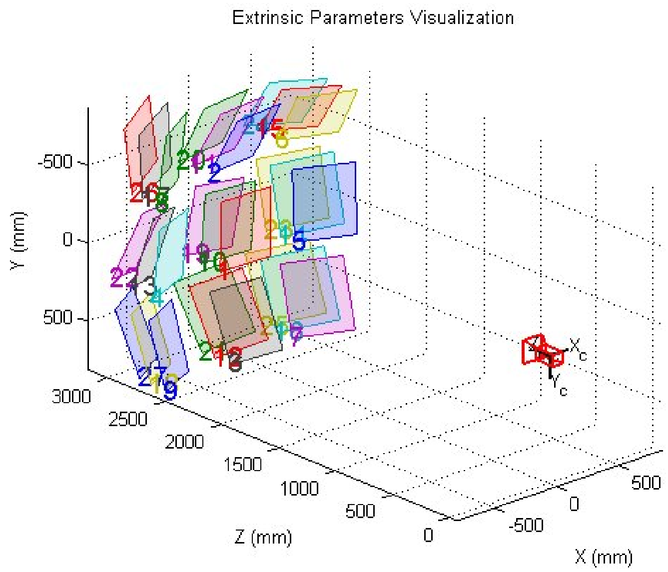


Figure 4.9: Arrangement of calibration patterns used for calibrating the camera in an example illustrating the effects of the distance of the calibration in the pinhole model under unfavourable parameter conditions.

The following simulation example is designed to illustrate that regardless of the setup parameters, it is possible to get a reasonable approximation of the physical state with a standard SVP pinhole model, if the observed part of the scene is always recorded from roughly the same distance D and the camera calibration was performed at about the same distance, i.e., the calibration pattern was moved underwater in front of the camera, roughly in the distance D as well. This illustrates that, e.g., for mosaicking with a vehicle camera in a (roughly) fixed distance over ground, good rectification results with a pinhole model can be achieved if the calibration pattern was moved in water at the roughly same distance. On the other hand, errors emerge once the camera is looking at parts of the scene that are closer or further then D.

The setup parameters are as follows: $d_0 = 80mm$, $d_1 = 20mm$, i.e., a significant amount of space between the focal point of the physical camera lens and the glass panel plus a relatively thick glass pane. So, the parameters, especially d_0, are in this case relatively unfavourable.

In this illustrating example, the camera is calibrated with 27 pattern poses spread around a point 2 m away from the camera (Fig. 4.9). Then test points are generated again randomly but around a given distance from the camera and, using the same method as above, projected onto the image plane. The reprojection error against the point distance to the camera is shown in Fig. 4.10.

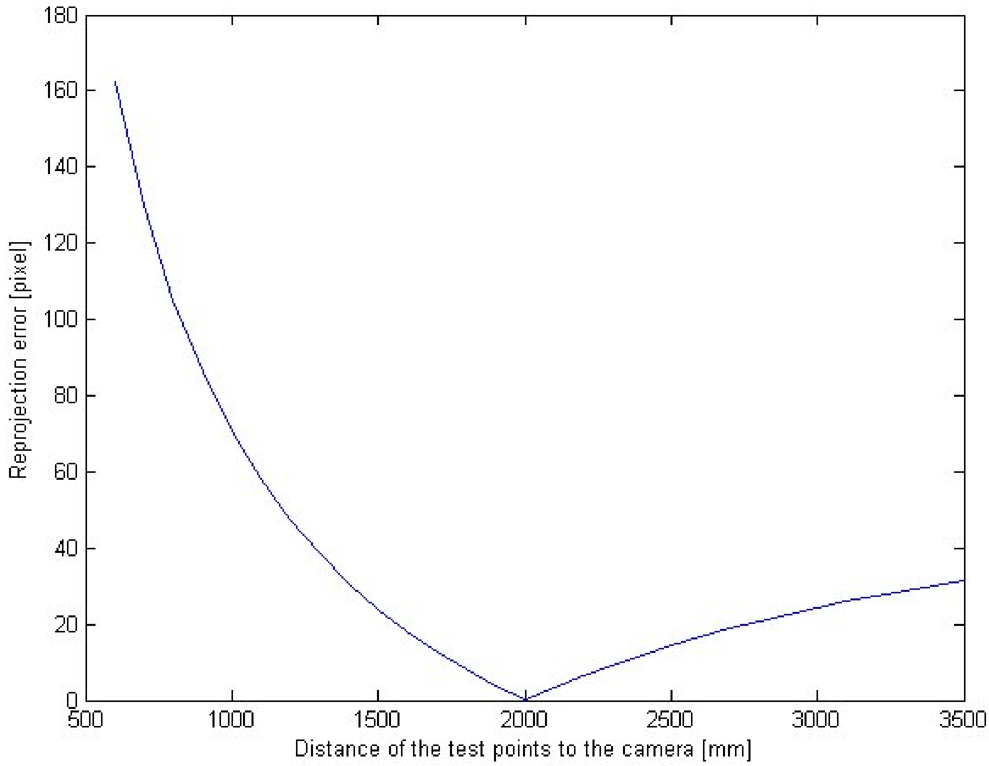


Figure 4.10: The reprojection error for a changing distance of observed points to the camera as an example that the SVP pinhole model performs well if the observed points are close to the distance in which the camera was calibrated with an SVP model. In this simulation example, the camera was calibrated with patterns around $D = 2$ m away from the camera (Fig.4.9), which is exactly the distance were the reprojection error is minimal.

The pinhole model holds very well only around the distance of calibration. This shows that for some specific applications, where minimizing d_0 is not possible, e.g., due to physical size of the lens as part of the camera subcomponent in the housing, the pinhole model can still be used effectively if the environment is observed from a known constant distance. As mentioned, one of the applications fulfilling this assumption can be seabed mosaicking with constant altitude control of the observing AUV.

This effect can also be observed in [Kang et al., 2012] where the quality of Structure from Motion under an SVP model is investigated and good results are reported even for a larger distance of the camera to the window. The camera rig used in the experiments leads to a constant distance between the camera and the investigated object; hence the effect illustrated in this section takes place.

Chapter 5

The Pinax Model

5.1 Overview

Based on the previous considerations, a system is proposed where a few setup assumptions are used to compensate for the refraction-based distortions of the image. Specifically, a transformation is computed to undistort and rectify the camera images. The resulting images can be directly used for example in stereo vision algorithms or for mosaicking, just to name two examples.

The following assumptions are made:

1. The distance d_0 between the glass and the centre of projection is small and near the optimal spot d_0^* where the rays traced back from the water cross in a minimum focus section Δx^*.

2. The optical axis of the camera is perpendicular to the glass surface. If not, a correcting transformation should be applied, e.g., for verged stereo systems (see section 5.4).

3. The glass thickness and its approximate refraction index are known, e.g., using standard refraction indexes for glass or plexiglass.

4. The water refraction index is approximately known from tables, e.g., from [Roswell et al., 1976].

Fulfilling these assumptions allows us to assume a pinhole model for the virtual camera and hence allows us to model the refraction-based distortions very efficiently. It also makes it possible to omit any underwater calibration procedures. The first assumption in the above list is of course by far the strongest and most significant one. As motivated before, it is at least not unrealistic to assume that underwater housings are minimized in size and that hence the physical camera inside the housing is placed as closely as possible to the window. This assumption is also supported by the real world experiments presented later on in Chapter 8.

5.2 In-Air Calibration

As the first step in our method, the physical camera ${}^{p}Cam$ is calibrated once in air, i.e., its intrinsic matrix $\mathbf{{}^{p}K}$ is determined using any standard calibration process [Hartley and Zisserman, 2003]. From a practical viewpoint it is very interesting to note that the front window does not have to be removed from the housing. To be more precise, the physical camera ${}^{p}Cam$ can be calibrated by calibrating the underwater camera ${}^{u}_{a}Cam$ in air, i.e., by determining its intrinsic matrix $\mathbf{{}^{u}_{a}K}$.

The air-glass and glass-air refractions only lead to a change in perceived camera location, which is part of the extrinsics, and the relative geometric relations between points in the scene are preserved. A calibration process of the underwater camera ${}^{u}_{a}Cam$ in air is hence the same as when calibrating ${}^{p}Cam$, i.e., $\mathbf{{}^{p}K} = \mathbf{{}^{u}_{a}K}$.

If the calibration of the physical camera was already done outside the housing, e.g., by the manufacturer, it is of course perfectly fine to use that data. The in-air calibration of the full underwater system is only an option that is very convenient to use for already existing complete camera systems. For high quality in-air calibration, the tool in CamOdoCal([Heng et al., 2013, Heng et al., 2014, Heng et al., 2015]) is used in our experiments presented later on in Chapter 8.

5.3 Determining the Optimal d_0^*

In an ideal scenario, the optimal distance d_0^* between the glass and the centre of projection can be taken into account when designing a new underwater camera. More precisely, the optimal distance ${}^{p}d_0^*$ of the physical camera should be taken into account, as the model of the virtual camera ${}^{v}Cam$ has its own, slightly different ${}^{v}d_0^*$ as discussed in more detail in the following section.

As already sketched in Sec. 4.3, ray tracing and non-linear optimization can be used to minimize the length of the section where light rays back-traced from the water intersect with the camera optical axis.

Using ${}^{p}d_0^*$ in a camera design is the ideal scenario and its computation is included here for the sake of completeness. In most application cases, the underwater camera is an off-the-shelf system or an already finished design. Other design constraints on the housing or the physical camera/lens components may apply as well. However we consider it safe to assume that for any typical underwater housing the real distance ${}^{p}\hat{d}_0$ is sufficiently close to ${}^{p}d_0^*$.

As illustrated in Tab. 5.1, ${}^{p}d_0^*$ tends to be in the order of a few millimetres and less. At the same time, the physical length of lenses tends to be in the order of their focal lengths, i.e., the centre of projection tends to be at the front-end of the lenses of the camera device. Hence placing the physical camera as close as possible to the glass-pane with possibly a small air gap, i.e., using the standard default set-up for typical underwater cameras, leads to a close approximation of ${}^{p}d_0^*$ by ${}^{p}\hat{d}_0$ with negligible errors. An exact quantification of the related errors is discussed below in the following sections.

Table 5.1: Optimal $d_0^*/^v d_0^*$ of the centres of projection of the physical/virtual camera for different glass thicknesses and two common salinity values

d_1 [mm]	$n_w = 1.333$(fresh water)	$n_w = 1.342$(salty water)
1	0.15mm/0.06mm	0.14mm/0.06mm
3	0.45mm/0.18mm	0.42mm/0.17mm
5	0.76mm/0.31mm	0.70mm/0.29mm
10	1.52mm/0.61mm	1.40mm/0.58mm
15	2.28mm/0.92mm	2.10mm/0.87mm
20	3.04mm/1.22mm	2.80mm/1.15mm

5.4 Verged Systems Behind Single Flat Panel

The assumptions made about the hardware setup allow to treat a physical camera as if it was in the optimal position with respect to the glass panel without causing significant errors. These assumptions and further knowledge of the intrinsic and extrinsic parameters of the complete physical camera system in air allow to compute the virtual intrinsic and extrinsic parameters for underwater usage. However, when using verged system behind a single glass panel assumptions presented in Sec. 5.1 must be modified. As the cameras are verged they cannot be orthogonal to the glass panel. Therefore the images must be corrected to rectify the image with respect to the glass panel first. The schematic view as applied to a general stereo camera is presented in Fig. 5.1.

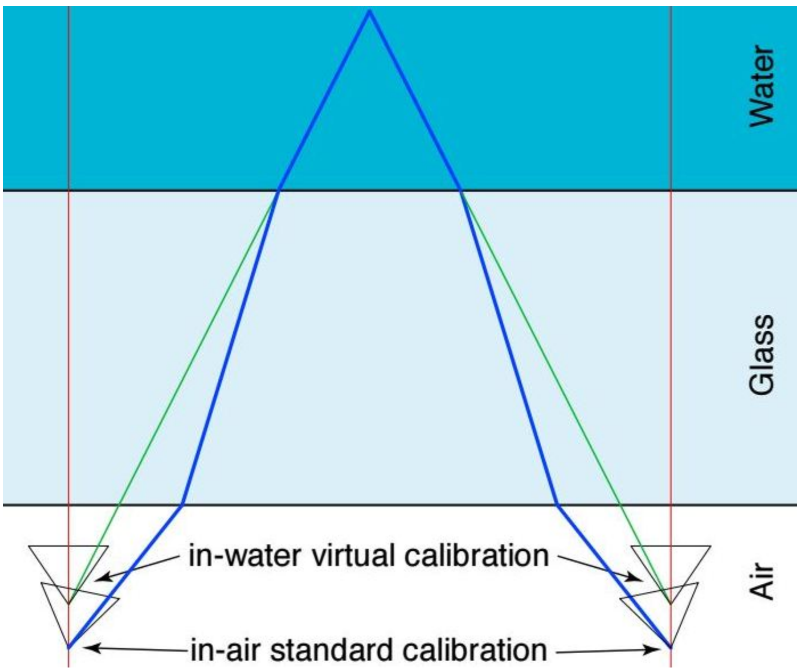


Figure 5.1: Real and virtual cameras used in the method.

First, the rotation between the two real and the two virtual cameras must be found ($\mathbf{R_{right}}$ and $\mathbf{R_{left}}$). To this end, both virtual cameras are defined to have orthogonal optical axes to the glass panel and the Pinax model may be used. However, there is a known rotation between these cameras $\mathbf{R}$ that can be determined by in-air calibration. The $\mathbf{x}$ axis of the final stereo camera system is defined to be along the translation vector $\mathbf{t}$ between the cameras. This ensures that the resulting camera $\mathbf{x}$ axes are aligned. The $\mathbf{z}$ axis is taken as the average of the two in-air camera optical axes. The $\mathbf{y}$ axis is set accordingly to the $\mathbf{x}$ and $\mathbf{z}$ axes. This defines the rotation matrices for both cameras used later for the image rectification:

$$
\begin{aligned}
\mathbf{x} &= -\frac{\mathbf{t}}{|\mathbf{t}|} \\
\mathbf{a} &= \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \mathbf{R}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\right] \cdot 0.5 \\
\mathbf{z}' &= \frac{\mathbf{a}}{|\mathbf{a}|} \\
\mathbf{y} &= \mathbf{z}' \times \mathbf{x} \\
\mathbf{z} &= \mathbf{x} \times \mathbf{y} \\
\mathbf{R_{right}} &= (\mathbf{x}, \mathbf{y}, \mathbf{z}) \\
\mathbf{R_{left}} &= \mathbf{R_{right}}\mathbf{R}^T
\end{aligned}
$$

Here, $\mathbf{R_{right}}$ is constructed by vector concatenation. The transformation $(\mathbf{t}, \mathbf{R})$ is assumed to describe the left camera coordinate frame relative to the right. This is equivalent to changing the original Pinax assumption, that the camera axis is orthogonal to the glass panel, to a new assumption: a baseline between the cameras must be parallel to the front glass panel. In most cases, it will be true as even when the cameras are verged they are usually mounted on the same plane for simpler construction. Once the rotation between the real and virtual cameras is known, the image correction can be computed.

Chapter 6

Modelling Refraction-based Distortions

6.1 Integrated Lens and Refraction-based Distortions Modelled with Maps

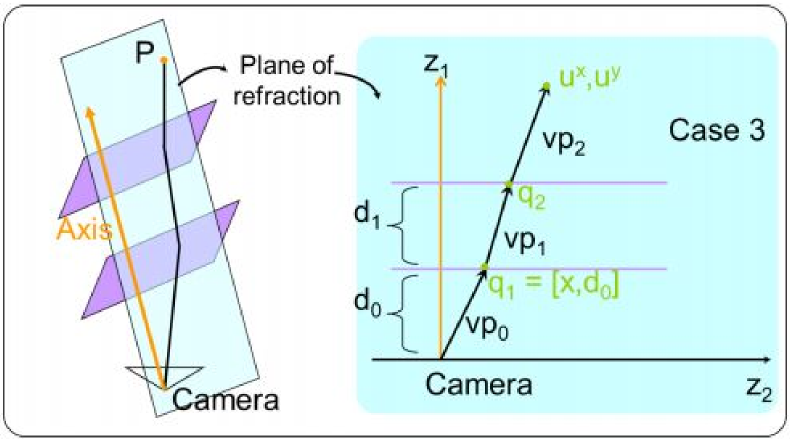


Figure 6.1: Parameters of the analytical forward projection through a flat refractive panel with a 12th degree polynomial (reprint from supplementary materials to [Agrawal et al., 2012]).

The main conclusion from the assumptions in Sec. 5.1, especially from assumption 1 about the distance of the physical camera to the window, is that a pinhole camera model can be used for the virtual camera model with a negligible error. To be more precise, this insight is exploited by defining a virtual Pinax plane $\mathbf{p_{pa}} = (d_{pa}, \mathbf{n_{pa}})$ that is assumed to be at distance d_{pa} in the scene with a normal vector $\mathbf{n_{pa}}$ that is parallel to the camera axis. The distance d_{pa} is set fixed to $5m$ as this is considered a typical viewing distance; but as discussed below, the exact value is of minor interest as points on Pinax planes at different distances behave similar due to the virtual pinhole camera property that follows from small values of d_0. $^{\mathbf{v}}\mathbf{p}$ and $^{\mathbf{p}}\mathbf{p}$ denote homogeneous image pixel coordinates. The intensity or colour value of a given pixel is denoted as $I(\mathbf{p})$.

Each point $^{\mathbf{v}}\mathbf{p}$ from the image ^{v}I of the virtual camera ^{v}Cam is projected onto the Pinax plane $\mathbf{p_{pa}}$ using a pinhole camera projection. Then this point $\mathbf{m_w}$ is projected forward to the inside surface of the glass panel (point $\mathbf{m_a}$) using the method derived from [Agrawal et al., 2012]. Now $\mathbf{m_a}$ may be transformed to $^{\mathbf{p}}\mathbf{p}$ with the in air calibration parameters of ^{p}Cam to obtain pixel coordinates in the distorted image ^{p}I. This last step may be performed using any camera and lens distortion model, referenced in Algorithm 1 with the subroutine *project3dToImage()*. When the calibration of the physical camera is based on a pinhole camera model with no lens distortion, this is:

$$^{\mathbf{p}}\mathbf{p} = {}^{\mathbf{p}}\mathbf{K} \cdot \mathbf{m_a}$$
$$^{\mathbf{p}}\mathbf{p} = {}^{\mathbf{p}}\mathbf{p} \cdot \tfrac{1}{^{p}p_z}$$

In order to find the point $\mathbf{q_1}$ (point corresponding to $\mathbf{m_a}$, expressed in coordinate frame z_1, z_2) as shown in figure 6.1 (compare also figure 4.1), the twelfth-degree polynomial method derived in [Agrawal et al., 2012] is used. For the sake of completeness the most important findings of [Agrawal et al., 2012] are presented here. As discussed before, it can be shown that a camera behind a flat glass panel is an axial camera. The camera axis is assumed to be identical to the optical axis. When tracing the light path all the refractions happen in one plane, called plane of refraction (POR), so the analysis can be conducted in 2D. To do this $\mathbf{m_w}$ must be projected to POR. The new coordinate system is defined as follows. Axis z_1 is identical to the camera axis, z_2 is orthogonal to z_1 and lays on POR. This way $\mathbf{m}_w$ projected to $\mathbf{u} = [u^x, u^y]$ may be used to derive the projection function (compare Fig. 6.1). This derivation starts with the constraint:

$$\mathbf{vp_2} \times (\mathbf{u} - \mathbf{q_2}) = 0$$

This constraint is valid as the refraction is analysed on the plane of refraction. Furthermore:

$$\mathbf{q_2} = \mathbf{q_1} - d_1\mathbf{vp_1}/\mathbf{vp_1^T n} = [x; d_0] - d_1\mathbf{vp_1}/\mathbf{vp_1^T n}$$
$$\mathbf{vp_2} = \tfrac{1}{n_g}\mathbf{vp_1} + b_2\mathbf{n} = \tfrac{1}{n_w}\mathbf{vp_0} + (\tfrac{n_g}{n_w}b_1 + b_2)\mathbf{n}$$

where:

$$b_1 = \left(d_0 - \sqrt{d_0^2 - (1-n_g^2)(x^2+d_0^2)}\right)/n_g$$

and:

$$b_2 = \frac{\sqrt{(n_g^2-1)(d_0^2+x^2)+d_0^2} - \sqrt{(n_g^2-1)(d_0^2+x^2)-(n_g^2-n_w^2)(d_0^2+x^2)+d_0^2}}{n_w}$$

Finally, substituting $\mathbf{q_2}$, $\mathbf{vp_2}$, b_1 and b_2 in the initial constraint we get:

$$k_1\sqrt{D_1} + k_2\sqrt{D_1 D_2} + k_3\sqrt{D_2} = 0$$

where

$$\begin{aligned}
k_1 &= x(d_0 + d_1 - u^y)\\
k_2 &= (u^x - x)\\
k_3 &= -d_1 x\\
D_1 &= d_0^2 n_g^2 + n_g^2 x^2 - x^2\\
D_2 &= d_0^2 n_w^2 + n_w^2 x^2 - x^2
\end{aligned}$$

Removing the square root terms it may be represented as:

$$(k_1^2 D_1 + k_3^2 D_2 - k_2^2 D_1 D_2)^2 - 4k_1^2 k_3^2 D_1 D_2 = 0.$$

Algorithm 1: Creating correction maps in the Pinax model (compare Fig. 6.2).

let M be an associative array
for ${}^v\mathbf{p} \in {}^vI$ **do**
  $\mathbf{m}_w = {}^vK^{-1} \cdot {}^v\mathbf{p} \cdot d_{pa}$
  $z_1 = (0,0,1)^T$
  $z_2 = z_1 \times (z_1 \times \mathbf{m}_w)$
  $u^x = z_2 \cdot \mathbf{m}_w$
  $u^y = z_1 \cdot \mathbf{m}_w$
  q_1=solve12thDegPoly(setupParams,$[u^x, u^y]$)
  $\mathbf{m}_a = q_{1\_x} \cdot z_2 + q_{1\_d} \cdot z_2$
  ${}^p\mathbf{p}$=project3dToImage(CameraAndLensModel,$\mathbf{m}_a$)
  store key-value pair $({}^v\mathbf{p}, {}^p\mathbf{p})$ in M
end

Algorithm 2: Applying Pinax correction maps

Let M be an associative container created with algorithm 1
for ${}^v\mathbf{p} \in {}^vI$ **do**
  look up value ${}^p\mathbf{p}$ for key ${}^v\mathbf{p}$ in M
  $\mathbf{a} = floor({}^p\mathbf{p})$
  $b_x = {}^p\mathbf{p}_x - \mathbf{a}_x$
  $b_y = {}^p\mathbf{p}_y - \mathbf{a}_y$
  $c_1 = b_x \cdot {}^pI(\mathbf{a}) + (1 - b_x) \cdot {}^pI(\mathbf{a} + (1,0,0)^T)$
  $c_2 = b_x \cdot {}^pI(\mathbf{a} + (0,1,0))^T + (1 - b_x) \cdot {}^pI(\mathbf{a} + (1,1,0)^T)$
  ${}^vI({}^v\mathbf{p}) = b_y \cdot c_1 + (1 - b_y) \cdot c_2$
end

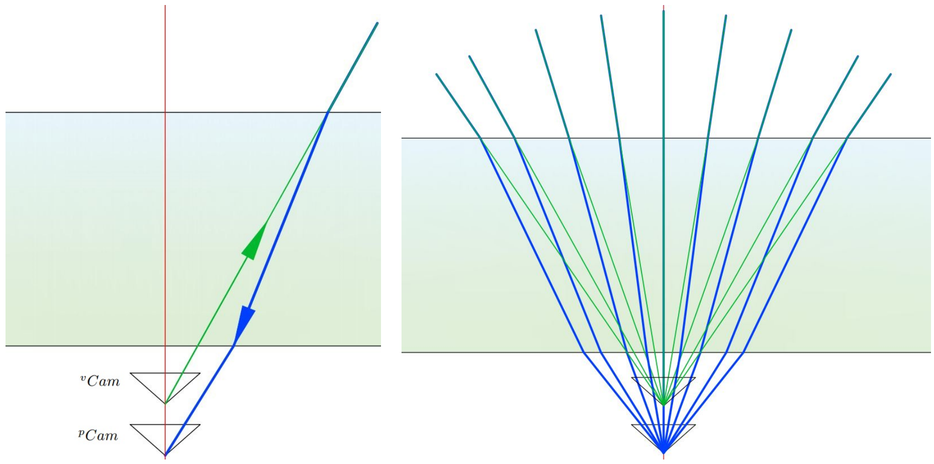


Figure 6.2: Left: The map creation in the Pinax model that combines a projection from the virtual pinhole camera to the Pinax plane (green ray) and back with an axial projection to the physical camera (blue ray). Right: The virtual (green) rays are good approximations of the physical rays (blue) once they cross from the glass panel into water - and the small d_0 assumption is fulfilled.

The method solving this polynomial to find q_1 is referenced with the subroutine *solve12thDegPoly()* in Algorithm 1.

This procedure that combines a pinhole forward and an axial backward projection has to be computed only once and leads to image transformation for undistortion and rectification stored in a lookup table (compare Algorithm 1 , 2 and Fig. 6.2). The main contribution in the context of the Pinax model is of course Algorithm 1, i.e., the way the correction map is created, while Algorithm 2 is just the standard procedure for applying correction maps for rectification, which is included here for the sake of completeness.

6.2 Refraction-based Distortions Modelled with Distortion Coefficients

As shown in the previous section the lens and refraction-based distortions may be combined into a single correction map. However it is also possible to model both sources of image deformation separately. Lens distortion correction map is usually encoded through a set of parameters dependent on the mathematical model that is being used. The same model may be used to encode the refraction correction. Therefore in this section it is assumed that the lens distortion will be corrected separately. Only refraction-based distortion is modelled, as an additional layer of image deformation (compare Fig. 6.3).

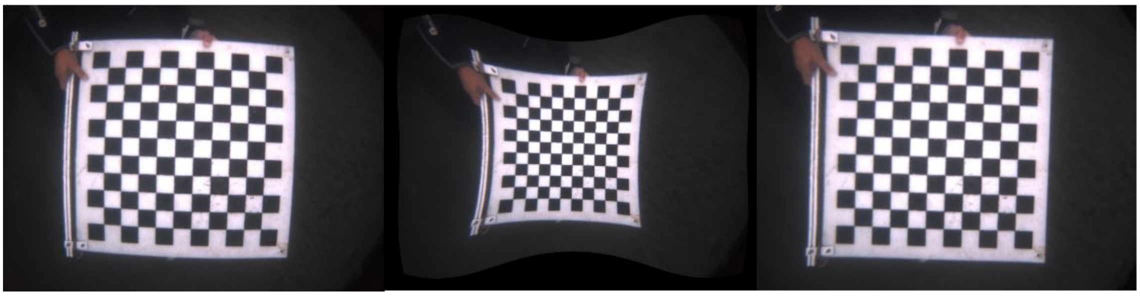


Figure 6.3: Correcting lens and refraction-based distortions separately. Left: raw image, centre: image after removing lens distortions, right: fully corrected image

6.2.1 Modelling method

The refraction effects on the image may be modelled with lens distortion model and change in focal length. This requires a few steps to be performed. Firstly lens distortion is corrected in the image. At this point any additional effects will be caused by refraction only. Then a grid of 300 arbitrary points covering the whole image ${}^{p}I$ is selected. For these points ray tracing is performed as shown in Sec. 4.2. This step results with a set of vectors corresponding to the light rays in the water. These vectors are then moved to the centre of projection of the virtual camera ${}^{v}Cam$ to simulate its image with a new set of points corresponding to initially selected set.(Fig. 6.4). Unlike the correction presented in Sec. 6.1 this method starts the analysis at the ${}^{p}Cam$ and traces the light to the ${}^{v}Cam$. This allows to omit projection with 12th degree polynomial. However after tracing points from ${}^{p}Cam$ to ${}^{v}Cam$ a transformation ${}^{p}I \rightarrow {}^{v}I$ still remains to be found. The goal is to find a set of distortion coefficients [Zhang, 2000] and apparent change in focal length that will transform initially selected set of points to their desired positions. For this a non linear optimization is performed. However only four distortion coefficients are optimized. In the error function the image is undistorted with the currently tested parameters. Then the new intrinsic matrix that best scales this points to their ideal position is found. It is done by matching the light ray vectors corresponding to this point and the ideal ones. These vectors may be easily calculated with:

$$\mathbf{v_0} = {}^{\mathbf{p}}\mathbf{K}^{-1}\,{}^{\mathbf{p}}\mathbf{p}$$

Where $\mathbf{v_0}$ is one of the vectors that should be modelled. Then the new intrinsic matrix may be found with least squares method, as follows:

$$\begin{aligned}
\mathbf{X} &= [p_{ux} \quad 1] \\
\mathbf{Y} &= [p_{uy} \quad 1] \\
\begin{bmatrix} fxin_{new} \\ uin_0 \end{bmatrix} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{v_{0x}} \\
\begin{bmatrix} fyin_{new} \\ vin_0 \end{bmatrix} &= (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{v_{0y}}
\end{aligned}$$

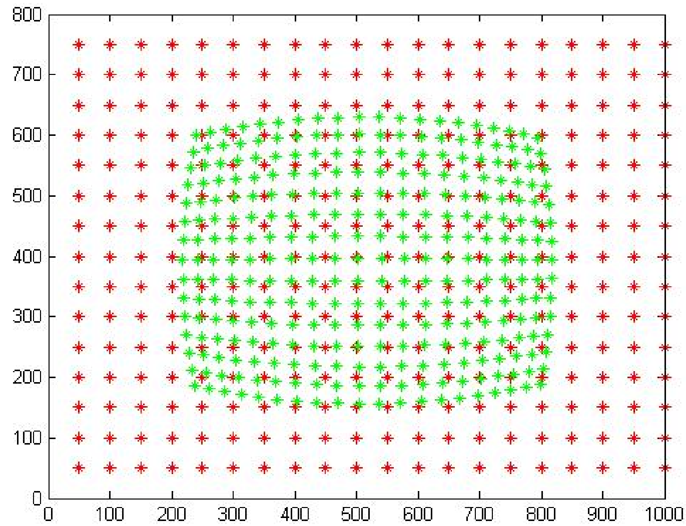


Figure 6.4: Set of arbitrary points of the image (red) and corresponding corrected positions (green). Please note that the registered image points should be moved into barrel-like shape to correct pincushion distortion caused by the refraction. This agrees with direct observation presented in Fig.6.3.

Where $\mathbf{p_u}$ are coordinates of the undistorted points and additional subscripts x or y means that only respective coordinates are taken into account. All parameters found this way create inverted matrix of the new intrinsic matrix so:

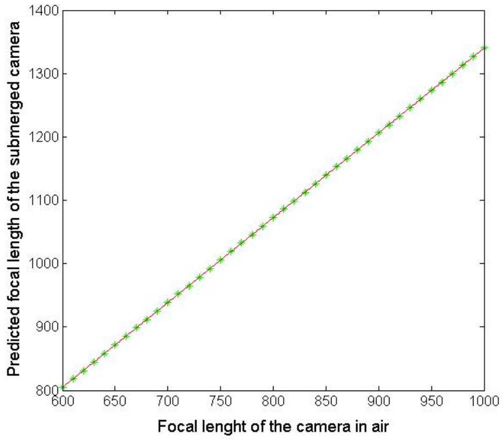
$$^{\mathbf{v}}\mathbf{K} = \begin{bmatrix} fxin_{new} & 0 & uin_0 \\ 0 & fyin_{new} & vin_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}$$

Difference between calculated light rays vectors (corresponding to current distortion coefficient and intrinsic matrix) and ideal vectors is then passed to the optimizing function as a parameter that should be minimized.

6.2.2 Modelling results

In this section all conditions are assumed to be known. The centre of projection is exactly in the spot, where the pinhole camera model is valid ($d_0 = 1.4282mm$), the glass surface is orthogonal to the camera optical axis and the refraction index of the water is known exactly ($n_w = 1.34$).

Then for this optimal parameters simulations were performed, as discussed in Sec. 6.2.1, to find the predicted values of the camera intrinsics and the distortion coefficients underwater. These simulations were repeated for different focal lengths to depict usage of different cameras being used in accordance to the method proposed here. The results are shown on Fig. 6.5.



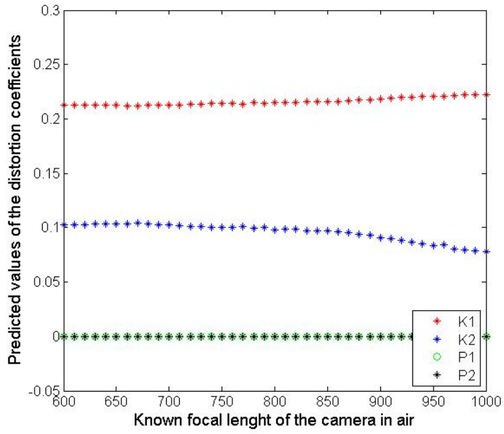


Figure 6.5: Left: Simulated values of the ${}^{v}Cam$ focal length depending on this value for ${}^{p}Cam$. Right: Predicted values of distortion coefficients required to model refraction-based distortions. Focal length in both is given in pixel units, as in the intrinsic matrix.

Fig. 6.5 shows the prediction of the focal length depending on the focal length of the input camera. The simulated data may be very well estimated with linear function:

$$f_{predicted} = 1.34 \cdot f_{input}$$

This result agrees with theoretical prediction formulated in [Lavest et al., 2003]:

$$f_{predicted} = \frac{n_w}{n_a} \cdot f_{input}$$

Furthermore, as visible on Fig. 6.5 both tangential coefficients are zero and the refraction may be entirely modelled with two radial coefficients only. What is even more important, these coefficients are nearly constant, any changes may be result of residual error in the optimization. Nonetheless both can also be estimated with linear functions:

$$\kappa_1 = 2.6517e - 05 \cdot f_{input} + 0.19468$$
$$\kappa_2 = -6.1319e - 05 \cdot f_{input} + 0.14491$$

Approach based on lens distortion model applied to refraction correction has an advantage of being more compact. Correction maps require significantly more space when being stored. On the other hand maps should be more precise as correction is calculated through analytical equations, not optimization. Furthermore image is corrected only once, lens and refraction-based distortions are being corrected in one go so interpolation of pixels' values is performed only once. Finally lens distortions tend to have barrel nature, where refraction-based are pincushion type. They compensate each other (compare Fig.6.3) which is yet another reason to use maps. Therefore in the rest of this thesis whenever Pinax model is used image correction based on correction maps is used.

Chapter 7

Numerical Error Analysis

In previous chapters a way of correcting refraction-based distortions with Pinax model was presented. This model and its limits will now be tested in simulations. Simulated camera is calibrated and different parameters are being changed, to estimate their influence on the model.

7.1 Influence of the Pinax Plane and d_0 Distances on the Correction Maps' Accuracy

The essential assumption in our model is that the correction computed for points in the Pinax plane also generalizes for other points in the scene. Furthermore, we postulate that minor variations in $\hat{d}_0$ are negligible and that typical underwater cameras are already designed in a way that allows to fulfil near optimal conditions. Fig. 7.1 shows the maximum errors between look up pixel values for the optimal $^*d_0$ and a Pinax plane distance of 5m and scene points that are at different distances than the Pinax plane, respectively if in addition d_0 deviates from the optimal $^*d_0$.

Fig. 7.1 shows that the errors are very small, i.e., in the order of at most a few millimetres over some meters distances, even with significant deviations of d_0 from $^*d_0$ of up to 40mm, i.e., under the presence of severe air gaps between the camera and the front panel. Only if the physical camera is significantly placed away from the glass panel pronounced errors occur. If d_0 is quite close $^*d_0$, i.e., if the air gap is small, the theoretical errors are even negligible considering realistic camera parameters. It can also be noted that the error becomes smaller for larger distances of the scene points.

7.2 Error Introduced by Camera-Glass Orthogonality Error

The second assumption presented in Sec.5.1 states, that the camera's optical axis is orthogonal to the front glass panel. In this section Pinax model is further tested in

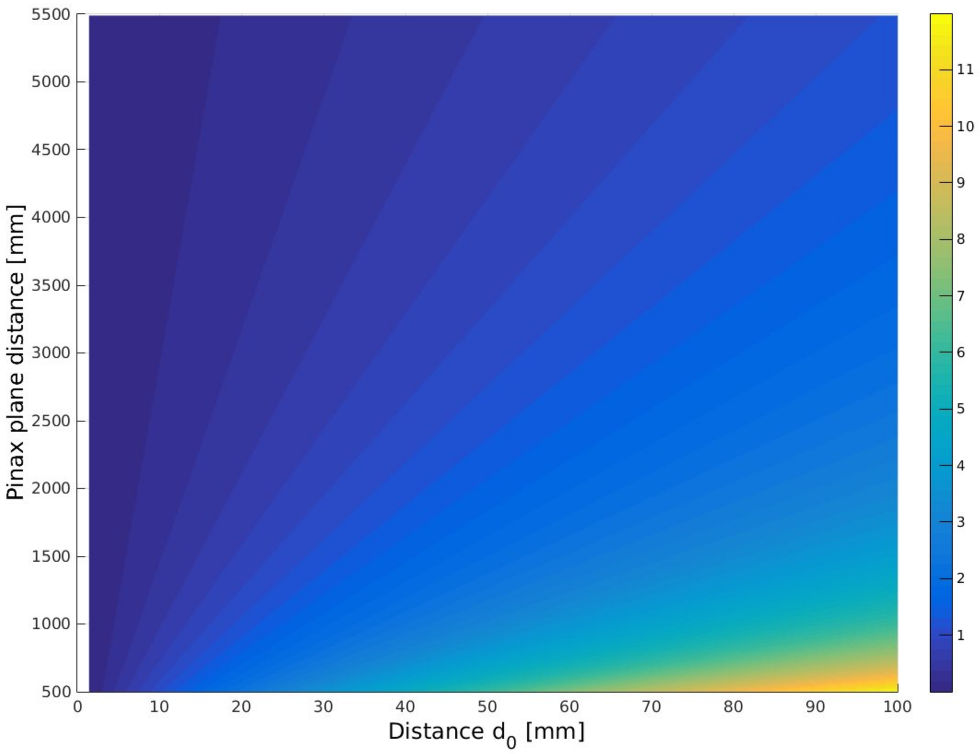


Figure 7.1: Errors between the look up pixel value for the optimal $^*d_0$ and the Pinax plane distance of 5m and scene points that are at different distances than the Pinax plane, respectively for which in addition d_0 deviates from the optimal $^*d_0$. Note that as long d_0 is close to $^*d_0$, the location of the point in the scene has no influence.

simulations, where the normal vector of the glass surface is rotated. This will lead to finding a maximum error introduced by this assumption.

The camera was modelled with the assumption that it is in its optimal position, $^*d_0$ away from the glass panel and with optical axis orthogonal to the glass. Then each iteration of the simulation moves rotates the camera up to $0.5°$. A set of arbitrary points covering whole image is traced into the water to the plane 5m away from the camera. Once the accurate model is used and, for the second time the Pinax model, assuming perfect orthogonality. Results of this experiment depicts Fig. 7.2. The horizontal axis represents angle between simulated normal vector and assumed one in degrees. The vertical axis represents average and maximum error found in the experiment.

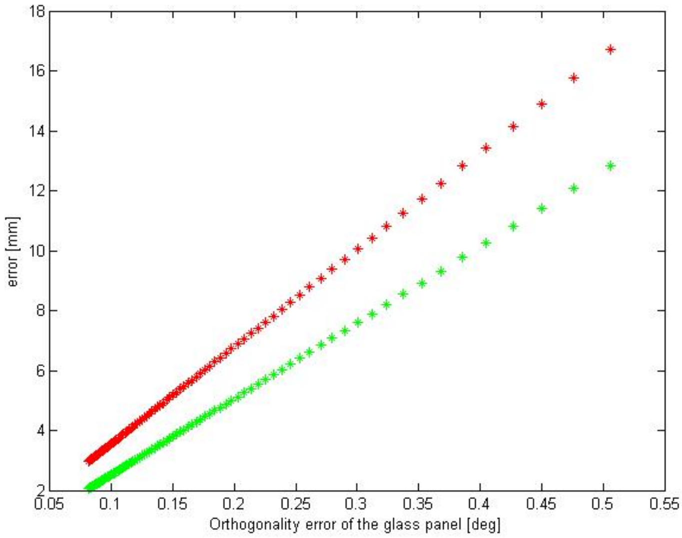


Figure 7.2: Maximum (red) and average (green) error, caused by orthogonality error. Error was measured as a displacement of the estimated light ray from the accurate one 5m away from the camera

The error grows linearly and for biggest assumed orthogonality error of the glass panel can reach 16.8mm max and 12.8mm on average. It is important to notice that the error is biggest for the points on the sides of the image. However these points are often not used in stereo reconstruction.

7.3 The Role of Changes in the Water Refraction Index

The computation of the predicted underwater camera parameters takes the refraction index of the water into account. Importance of this factor was tested in simulations. Similarly to the experiment in Sec.7.2 underwater camera is modelled with

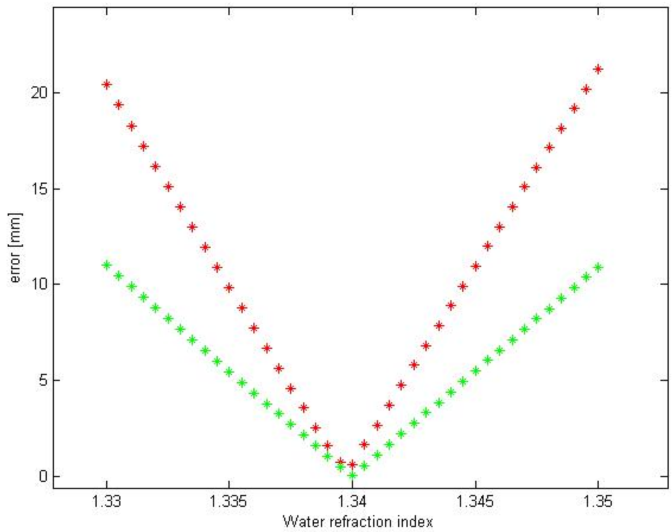


Figure 7.3: The maximum (red) and average (green) error caused by changes of the water refraction index in an example camera set-up. The error is measured as the displacement of the estimated light ray from the proper one at a 5 m distance from the camera.

Pinax method. Water refraction index $n_w = 1.34$ is assumed. Then all the setup parameters are set to the ideal values, only water refraction index is varied within the range $n_w = 1.33 - 1.35$. For each value of the water refraction index the physical model is used to trace light rays, corresponding to the set of points covering the whole image, into to water to the plane 5m away from the camera. Displacement between these rays and the ones predicted with Pinax model is taken as an error measure. Both the maximum error, which occurs for rays under the maximum incident angle, as well as the average error over all rays are shown. The errors are substantial, i.e., though there is only a very small change in the water refraction index, it is very beneficial to take it into account. This also holds for other methods in general as shown in the experimental results section. While it is sufficient to simply recompute the correction map in the Pinax model, which can be done very fast and without the need of gathering any additional vision data, the standard in water calibration approach requires a new recording of in-situ data to avoid errors.

In our experiments presented below, we simply use estimated salinity and the related refraction indexes from tables [Roswell et al., 1976], which we found sufficiently precise to accommodate for the effects of changing water refraction indexes. Nevertheless, the exact water refraction index can also be computed from physical parameters, e.g., by using the formulas from [Millard and Seaver, 1990] or [Quan and Fry, 1995]. The predominant factor is the influence of salinity followed, to a much lesser extent, by temperature and pressure. Very commonly used CTD sensors provide exactly this information, i.e., it is very simple to get an exact indirect measurement of the water refraction index if needed. As the computation of the correction map is relatively fast (see Sec.8.1), this even allows an semi-online re-computation of the correction map during the mission if the conditions change, e.g., if the camera on a vehicle operating at sea passes an fresh water inflow, or if a mission ranges from warmer shallow waters to much colder deep waters.

7.4 Combined Errors

In this section the combined errors caused by the most important factors discussed before (non-optimal d_0, not orthogonal glass plane and water refraction index known only approximately) are investigated. Firstly it was checked, how two of those factors influence the average and maximum error, when occur at the same time. The error is defined as previously - as a displacement from the optimal position on the plane 5m away from the camera. Results are shown on the Fig. 7.4 and Fig. 7.5.

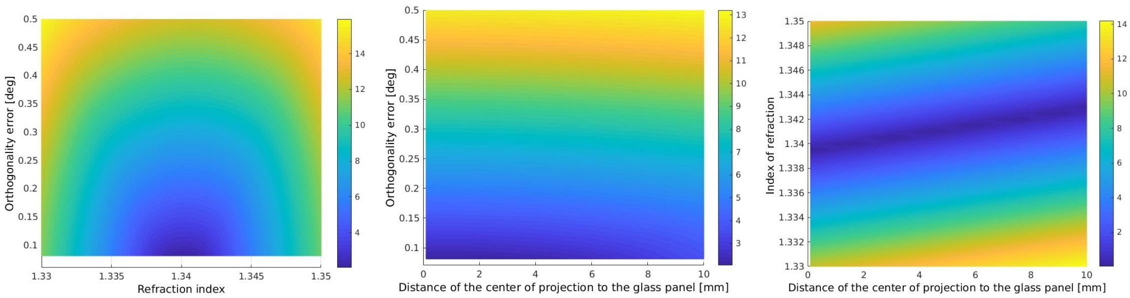


Figure 7.4: An average error, in mm, caused by the different pairs of the analysed error driving factors.

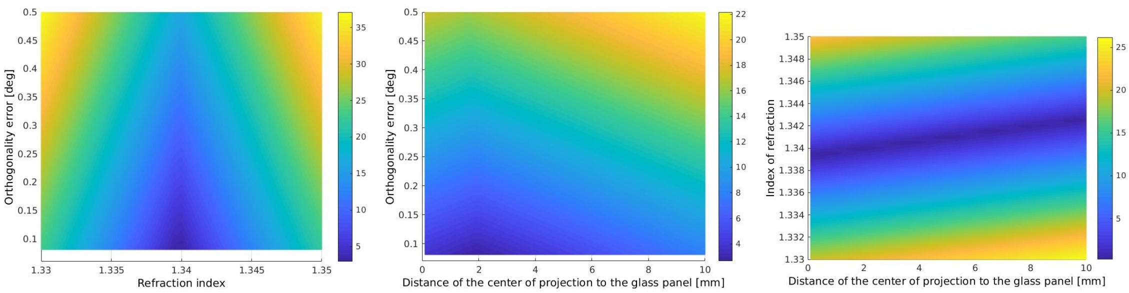


Figure 7.5: A maximum error, in mm, caused by the different pairs of the analysed error driving factors.

Analysing this plots it can be concluded that the maximum error will occur, when all discussed factors are at its assumed limits: the glass panel orthogonality error is 0.5°, water refraction index is smaller, than assumed and equals 1.33, and actual distance o the centre of projection to the glass panel reaches its maximum allowed value. For such conditions the average error equals 18.13 mm and maximum error is 42.84 mm. Of course this error will be smaller closer to the camera and bigger, when observed further than 5m away. It is worth noticing, that water refraction index bigger than assumed is compensated by increased d_0. It is also very important to notice, that in many cases the most important error driving factor - water refraction index- can be checked in the tables ([Roswell et al., 1976]) before the mission which will significantly decrease both maximum and average error, up to 22 mm and 13 mm respectively. Finally it must be noted, that most significant errors occur on the side of the image. However during 3D stereo reconstruction parts of the image on the sides are rejected as only scene seen by both, left and right camera, may be reconstructed. Therefore in practical setup measured erros may be smaller than predicted in these simulations.

Chapter 8

Experiments and Results

In this section the previous theoretical discussions and numerical analyses of the Pinax model are complemented with quantitative evaluations with real underwater camera systems. The underwater systems are based on various components received from different third parties. None of the systems or system components were designed with knowledge of the Pinax model. In all the experiments stereo cameras are being used. It allows for quantitative measurements to reliably asses the accuracy of the Pinax model. However monocular image may be corrected with the Pinax model just as easily and reliably as stereo pair.

Firstly, run-times for generating and applying the refraction correction maps are presented in Sec. 8.1. Especially the use of a correction map for rectification is extremely fast and can hence be applied in real-time on a video stream. Secondly, qualitative results from field work where the Pinax model is used for in-air calibration are presented in Sec. 8.2. The qualitative results are based on several third party systems including a custom-made underwater camera on the Ifremer vehicle Vortex [Brignone et al., 2011] and several COTS cameras in custom-made underwater housings, e.g., on the AUV Sparus [Mallios et al., 2011] from the University of Girona and the AUV Seacat [Enchev et al., 2015] from ATLAS Elektronik. Thirdly, quantitative evaluations are presented in Sec. 8.3 where the Pinax model is compared to state-of-the-art underwater calibration. The experiments are conducted with a Bumblebee XB3 with dual stereo, i.e., three monocular cameras at two different baselines, in a custom-made underwater housing and with a GoPro Hero3+ Black Edition stereo rig in a consumer housing from GoPro. The accuracy of underwater stereo computations on artificial checker-board patterns is used in the quantitative evaluations as a metric of rectification accuracy.

8.1 Run-Times for Generating and Applying the Refraction Correction Maps

One of the strengths of the Pinax model is its computational efficiency. The refraction correction maybe done via maps, i.e., simple look-up tables for image rectification which lead to very efficient operations very well suited for real-time performance. The computations of the maps themselves are also relatively fast and can be done just once offline. The following runtimes are benchmarked on an Intel Core i7-3610QM CPU running at 2.3 GHz, i.e., a mobile CPU that is used in an embedded system suited for integration on robotic vehicles, even within the camera system itself.

The experiments are done once with MATLAB R2014a on Windows 7 and once with the Robot Operating System (ROS) Hydro on Ubuntu 12.04. It should be noted that no optimization is used, and especially that no parallelization is employed. Both the computation of the correction map itself as well as its application for rectification can be easily speeded up by parallel computation, e.g., through multithreaded or CUDA programming if higher processing speeds are required.

Table 8.1: Computation times for generating the correction maps.

camera	resolution	time MATLAB (h:mm:ss)	time ROS (mm:ss)
Bumblebee2	1024x768	0:11:47	0:20
Bumblebee XB3 / Vortex cam	1280x960	0:18:25	0:32
GoPro Hero3+ Black Ed.	4096x2160	2:12:36	3:50
Fuji FinePix 3DW3	3648x2736	2:29:35	4:20

Tab. 8.1 shows the computation times of the correction maps for different cameras, respectively image resolutions. The computation is linear in the number of pixels and takes about 0.89925 msec/pixel on MATLAB, respectively 0.026042 msec/pixel on ROS. The computation of a Pinax correction map has to be done only once. It can hence be simply computed offline.

Each Pinax correction map depends - in addition to the in-air calibration map of the underlying physical camera - on the water refraction index, i.e., especially salinity. In the experiments reported later on, only two different correction maps are used across a wide range of different field experiments, namely one for salty water and one for fresh water. As discussed in more detail in the relevant sections, we found two maps to be sufficient. For even more accurate image rectifications, as mentioned earlier, it is possible to use a CTD sensor to determine the salinity of the water directly at the location of the mission, respectively even during the mission if the salinity changes. This then allows to either instantaneously switch between several pre-computed maps, or to even compute a perfectly fitting map online, which takes in the order of at most a few minutes under ROS, when no parallelization is applied.

Tab. 8.2 shows the computation times to apply the correction maps, i.e., to perform an image rectification, for different cameras, respectively image resolutions. The computation is just a look-up operation and hence very fast and very well suited for real-time operation. The underlying algorithm is again also well suited

Table 8.2: Computation times for applying the correction maps, i.e., for rectification.

camera	resolution	time MATLAB (seconds)	time ROS (seconds)
Bumblebee2	1024x768	0.025	0.007
Bumblebee XB3 / Vortex cam	1280x960	0.055	0.012
GoPro Hero3+ Black Ed.	4096x2160	0.412	0.085
Fuji FinePix 3DW3	3648x2736	0.453	0.094

for parallel computation; hence it is easy to further speed it up if necessary through multithreading or CUDA programming.

8.2 Qualitative Results

Table 8.3: The different camera systems used in qualitative evaluations

	camera	focal length (mm)	imager resolution	imager size	housing provider
1	Bumblebee XB3 (Jacobs)	3.8	1280x960	1/3¨	U.Zagreb
2	Bumblebee XB3 (IST)	3.8	1280x960	1/3¨	U.Zagreb
3	Bumblebee2 (UdG)	2.5	1024x768	1/3¨	UdG
4	Bumblebee2 (Jacobs)	2.5	1024x768	1/3¨	ATLAS
5	Vortex Camera	4	1280x960	1/3¨	Ifremer
6	Fuji FinePix 3DW3	6.3	3648x2736	1/2.3¨	FantaSea
7	GoPro Hero3+ Black Ed.	2.65	4096x2160	1/2.3¨	GoPro

In this section qualitative examples, covering a range of different underwater conditions, are given. They illustrate the usefulness of the method introduced in this Part for real world applications. Tab. 8.3 gives an overview of seven different systems where the Pinax model was used, i.e., the cameras in each system were calibrated just once in air and the Pinax correction tables were used for rectification of the images. The correction tables were computed with two different refraction indices, namely $n_w = 1.333$ for fresh water, $n_w = 1.342$ for salty water, respectively. Depending on the environment conditions, e.g., experiments in a pool or lake, or in the sea, the relevant map was chosen. The cameras have different technical parameters, especially with respect to focal length or d_1, and they are mounted in different housings that were all designed by third parties without any knowledge of the Pinax model, yet fulfilling the underlying assumptions. It was possible as these assumptions are convergent with usual requirements for such housings.

The test systems are all stereo cameras. The advantage of stereo cameras is in this context that they not only provide metric information, which will be used for a quantitative analysis later on, but that their data also provides very good qualitative indicators of the calibration and rectification accuracy. Stereo processing is very sensitive to the accuracy of the image rectification due to the inherent use of the epipolar constraint. If there are distortions in the two cameras, matching pixel blocks do not lie on the same line in the two images any more, i.e., the epipolar constraint is violated and correspondences cannot be established leading to missing range values. Hence rectification errors not only lead to metric errors in the range estimates but also to complete failures in the stereo computations.

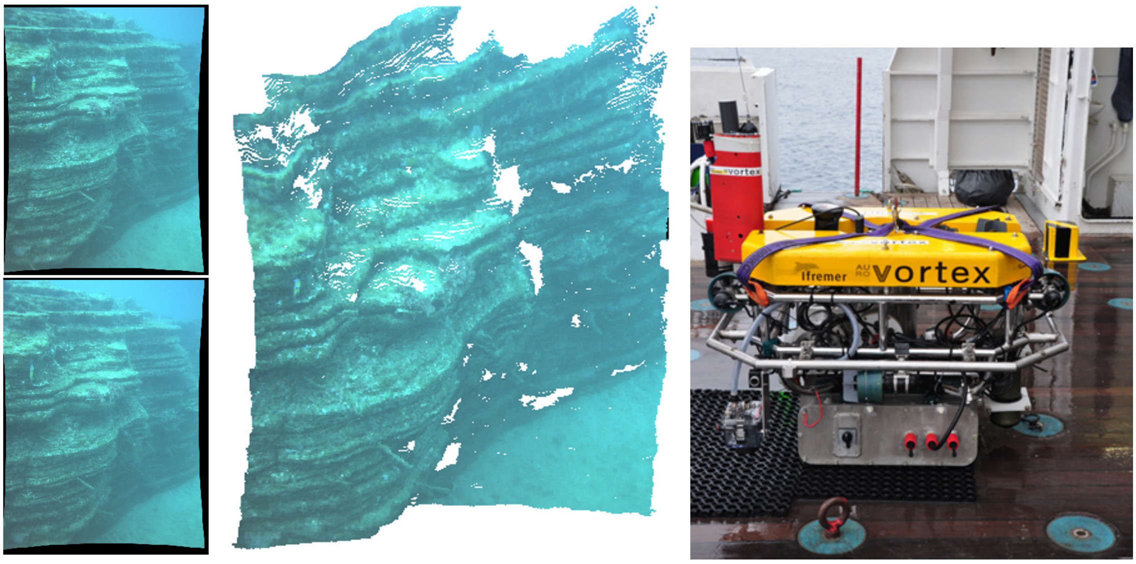


Figure 8.1: A 2.5D coloured point cloud (center) generated from images (left) from a custom-made underwater stereo camera on the Vortex vehicle (right) of the Institut francais de recherche pour l'exploitation de la mer (Ifremer). The stereo point cloud is very dense, hence indicating very good rectification accuracy.

The trials with different camera systems in various environment conditions show following main three qualitative results that are interesting for applying the method introduced in this article in real world applications:

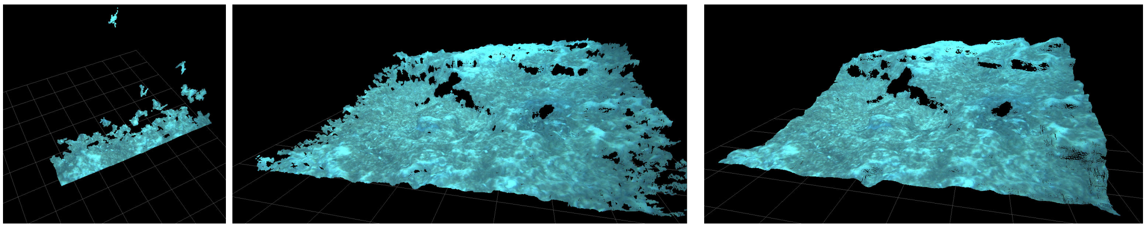
1. *In-air calibration of underwater cameras with the Pinax model is applicable to a range of systems and environment conditions.* Pinax-based calibration was applied to seven different systems used in different environment conditions. The cameras and housings were from various 3rd parties. In each case, in-air calibration with the Pinax model was successful and led to (at least) qualitatively comparable results to underwater calibration which was the previous state of practice for the systems.

2. *The quality of the in-air calibration matters.* The Pinax model allows for convenient in-air calibration that only has to be done once. The final result of the rectification is significantly influenced by the quality of this calibration.

Figure 8.2: Top: view from above, bottom: view from the side. A 3D map generated from 350 stereo based point clouds from a Bumblebee camera on a Sparus vehicle of the University of Girona. The consistently dense stereo data in all stereo computations indicates high rectification accuracy.

3. *The water refraction index, especially due to salinity, matters but to a lesser extent.* Ignoring the influence of the changes in refraction of water due to environmental parameters, especially in form of salinity, leads to degradation in accuracy in the rectification.

Regarding aspect 1., the Pinax model was successfully used on all seven systems. The in-air calibration and the related image rectification lead to high quality results in all cases as indicated by the density of the 3D point clouds generated by the stereo processing. Fig. 8.1 and 8.2 show two typical results as illustrative examples. It should be noted that the "holes" in the point cloud shown in the centre of Fig. 8.1 are visible just due to the perspective view, i.e., due to occlusions in the scene. In addition to the density of the stereo results, there are also qualitative indications of the metric accuracy.



(a) wrong in-air (b) wrong salinity (c) correct in-air and salinity

Figure 8.3: An illustration of two of the three main qualitative observations related to our method, namely that *2.* the quality of the in-air calibration matters (a) compared to c)) and that *3.* the salinity has an influence (b) compared to c)).

Fig. 8.3 illustrates the aspects *2.* and *3.* with respect to the relevance of the quality of the in-air calibration, depending on the water refraction index. The point cloud PC_c shown on the right was generated in seawater by system 2 (Bumblebee XB3 (IST) with U.Zagreb housing) using our method with the proper factory in-air calibration file as input and our standard salt-water refraction estimation. The resulting point cloud density ∇PC_c provides a comparison baseline for a simple illustrative example.

The point cloud PC_a shown on the left uses the factory in-air calibration file from exactly the same type of camera, namely the Bumblebee XB3 owned by Jacobs with identical (nominal) parameters as the one owned by IST, and which is mounted in the same type of housing, namely the design by U.Zagreb. The proper salt-water refraction index is used. Nevertheless, the point cloud density ∇PC_a is just 19.7% of the density ∇PC_c. So, no correspondences can be found for a significant portion of the pixels in both images, i.e., the necessary epipolar constraint for stereo vision does not hold, and so the rectification process is highly unsuccessful in this case.

The point cloud PC_b shown in the centre uses the correct factory in-air calibration file of this specific camera instance. However our standard fresh-water refraction index is here used in the Pinax model though the data is collected in seawater. The point cloud density ∇PC_b degrades therefore to 93.2% of the density ∇PC_c in this example. It can be noticed that there is especially missing data at the sides of the point cloud, which is consistent with what is to be expected when the rectification quality degrades. The distortion effects due to refraction are most visible at the sides of the stereo images, hence violations of the epipolar constraint due to degraded rectification start taking effect from there. Further examples of the Pinax model applications are given in the Chapter 16.

8.3 Quantitative Evaluation of the Pinax Accuracy

The numerical analyses of the Pinax model as well as the qualitative experiences in the field indicate that it leads to very accurate calibration and rectification results. This is now further substantiated with quantitative evaluations of real cameras, namely a Bumblebee XB3 (Tab. 8.3, system 1) and a stereo rig consisting of two GoPro Hero3+ Black Edition (Tab. 8.3, system 7). Both systems are quite different and provide two interesting test cases.

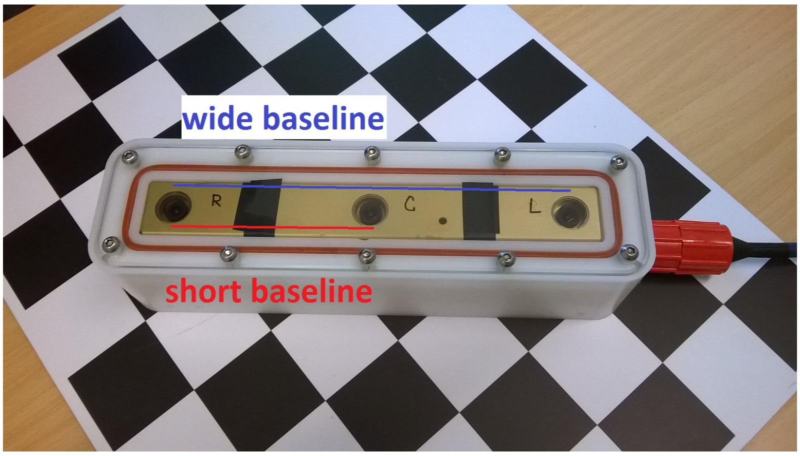


Figure 8.4: The Point Grey Bumblebee XB3 has three monocular cameras that allow stereo processing with a short and with a wide baseline. The chequerboard pattern underneath the camera is used for the quantitative accuracy analysis.

The Bumblebee XB3 features three monocular cameras. This allows stereo processing with a short and with a wide baseline (Fig. 8.4). The GoPro stereo system consist of a standard set-up with two cameras (Fig. 8.5). There are hence five monocular cameras in total that are calibrated and rectified with the Pinax model in the following experiments.

As it is difficult or even impossible to acquire ground truth data of natural underwater environments, the analysis is based on artificial chequerboard patterns where the exact distance between the black and the white fields is known. For the quantitative evaluations, the stereo systems are placed in a pool filled with fresh or

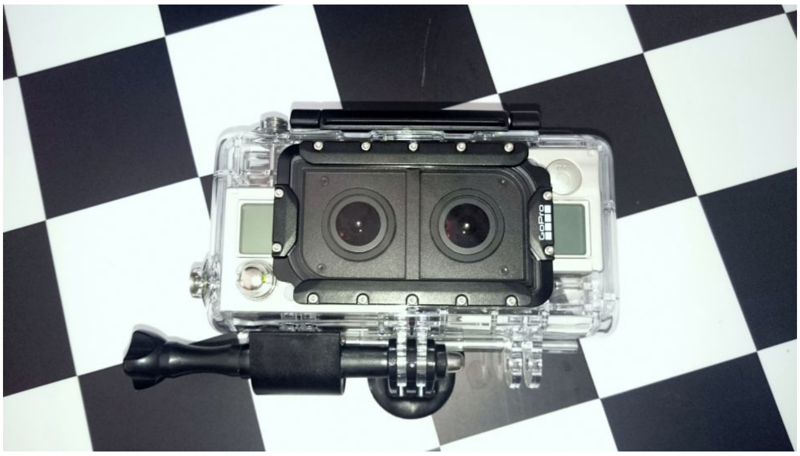


Figure 8.5: The stereo system consisting of two Gopro Hero3+ Black Edition cameras in a Gopro Dual HERO underwater housing.

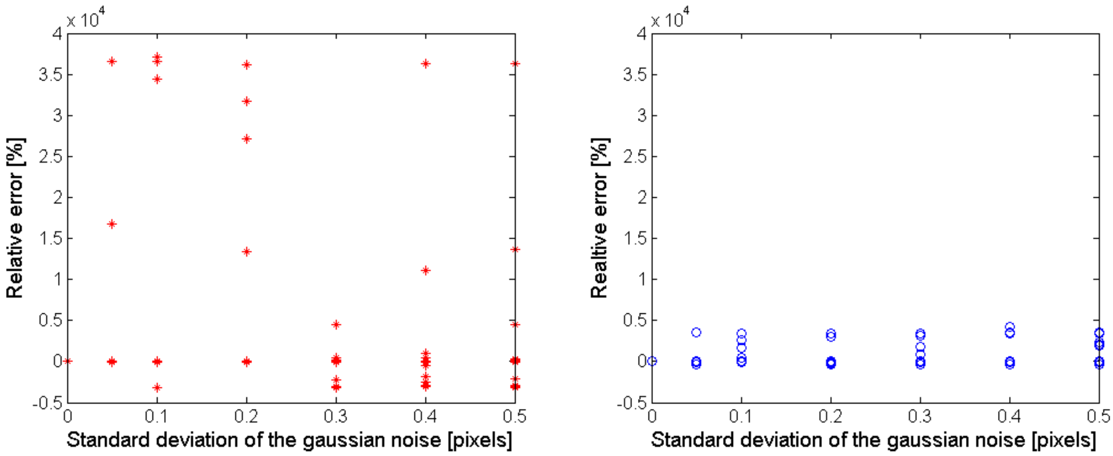
salty water. The chequerboard pattern is then moved at different distances within the field of view of each camera. Stereo processing is conducted for each sequence of images acquired at the different distances. The metric stereo estimates of the distances between the chequerboard markers are finally compared to the ground truth distances, thus providing an error metric for the rectification accuracy.

Four different methods for calibration and rectification are evaluated, namely:

- standard *in-air* calibration and rectification
- state-of-the-art *underwater* calibration with a *correct* water refraction index (WRI), i.e., the calibration is performed in-situ in water at exactly the same salinity conditions as the recording of the evaluation data that is then rectified
- state-of-the-art *underwater* calibration with a *wrong* WRI, i.e., the calibration is performed in a fresh water pool while the recording of the rectified evaluation data is done in salty water
- *Pinax* in-air calibration and rectification under arbitrary but roughly known (fresh or salty) water conditions

All calibrations are based on the popular SVP model by [Zhang, 2000] that can be found for example in MATLAB. We use here the according method from the CamOdoCal calibration package [Heng et al., 2013, Heng et al., 2014, Heng et al., 2015].

Pinax results unfortunately cannot be directly compared with the method of [Agrawal et al., 2012] or methods that build on it, i.e., that use the proper axial camera model to determine d_0 like for example [Kawahara et al., 2013]. The problem is that in these methods the non-linear optimization suffers from numerical instability when d_0 becomes small, i.e., when d_0 is in the order of several millimetres. More precisely, the underlying numerical optimization still performs well in these



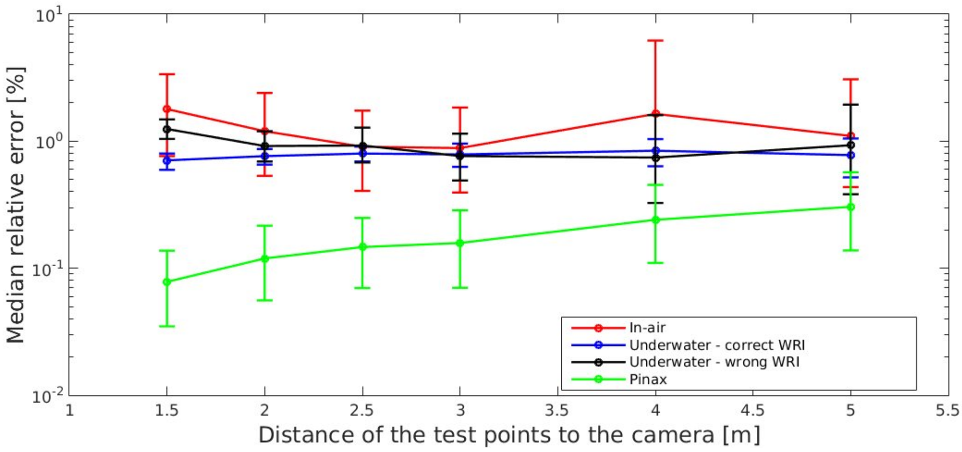
(a) d_0 1.5mm (b) d_0 15mm

Figure 8.6: The method from [Agrawal et al., 2012] becomes numerically unstable for small d_0 under the presence of noise. This is illustrated here with two datasets for $d_0 = 1.5mm$ and $d_0 = 15mm$. In each case, six noise levels from 0 to 0.5 pixel standard deviation are investigated. For each noise level, 15 runs of the method are performed on simulated data, i.e., with known ground truth d_0.

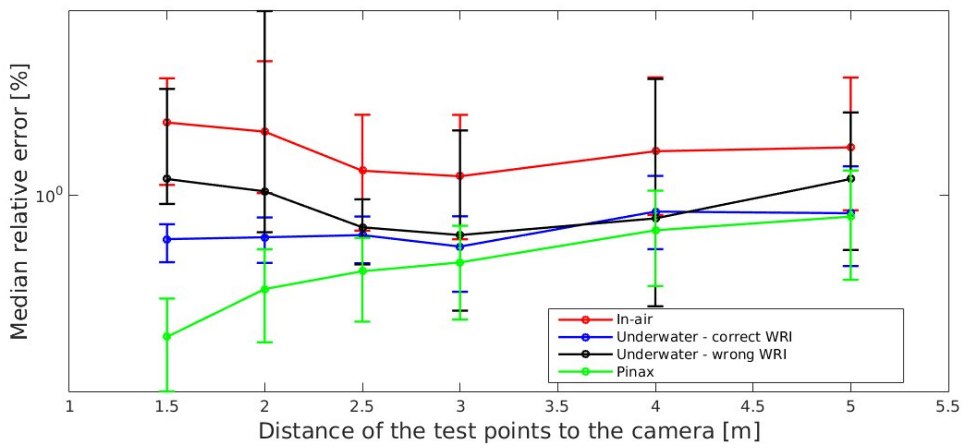
cases with perfect calibration data. However it diverges as soon as there is noise on the calibration data, especially when using any real world calibration data. It is interesting to note that the real world experiments presented in [Agrawal et al., 2012] or in publications building on it use quite large values for d_0, i.e., the cameras are placed at quite some distance, even half a meter away from the glass panel.

So in the region of interest for Pinax method, i.e., with the camera being close to the window and d_0 being in the order of a few millimetres, the exact, axial, camera model fails - the observed case starts to be outside the domain of the axial model. To substantiate this observation, experiments have been conducted with simulated data for different values of d_0, namely for $d_0 = 1.5mm$ and $d_0 = 15mm$. Simulated chequerboard points are projected from 3D to the image plane using the 12th degree polynomial for a proper axial camera model. Six levels of Gaussian noise from 0 to 0.5 pixel standard deviation are added to the projection. These points are used as input to the method from [Agrawal et al., 2012] using their released code. 15 runs are performed for each noise level for each value of d_0. The results of each run are shown in Fig. 8.6 as relative error of the result of the non-linear optimization over the ground truth. It can be observed that the method only leads to stable, correct results in the absolutely noise-free cases. For $d_0 = 1.5mm$, the output varies significantly over each run and produces significant errors in the estimate of d_0 as soon as there is the slightest noise present. It can be observed that the situation improves a bit for $d_0 = 15mm$, but a substantial instability still remains.

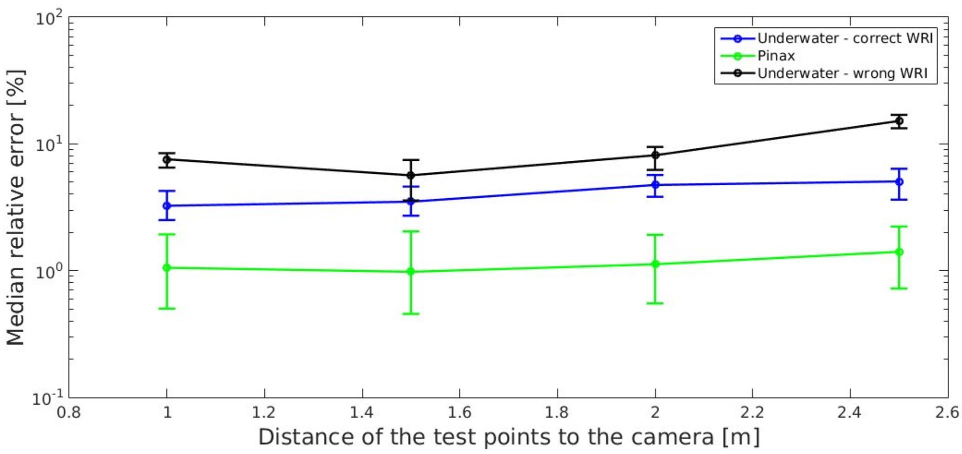
Fig. 8.7 shows the results of the evaluations on the three different stereo setups. All errors are normalized, i.e., they are plotted as percentage of the known,



(a) Bumble XB3, wide baseline



(b) Bumble XB3, short baseline



(c) GoPro

Figure 8.7: The relative errors of the triangulated points in % for the four evaluated methods on the three different stereo set-ups. The error bars show the upper and lower quartile values of the error, the centre dots are the medians. The error values are plotted for each nominal distance of the calibration pattern during the test.

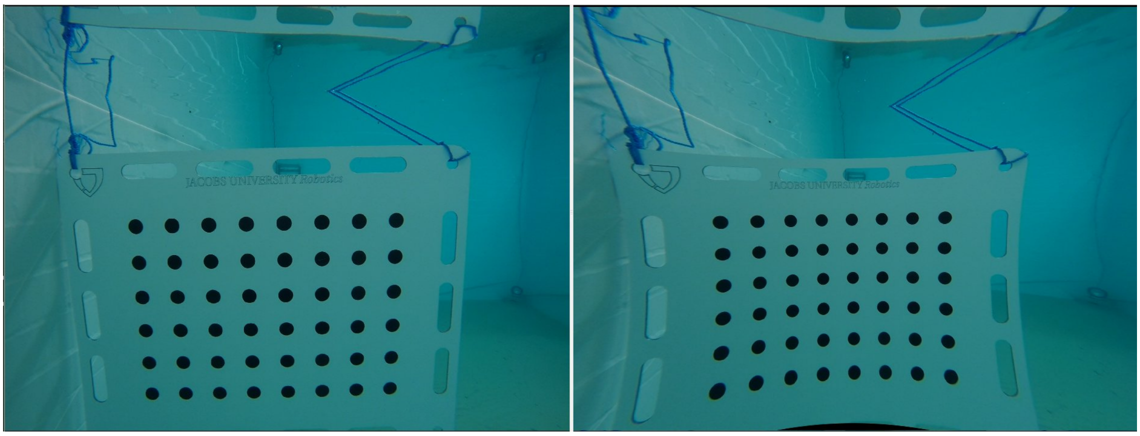


Figure 8.8: For the GoPro data, the in-air calibration leads to such severe distortions that the stereo processing is completely failing in this case. On the left, an example GoPro image from a test sequence is shown; on the right, the "corrected" image based on in-air calibration is shown.

measured distance between chequerboard markers. For the GoPro test sequence, the evaluation of the in-air calibration is omitted as its rectification is performing so poorly that stereo processing is not possible any more (Fig. 8.8). Also in the case of the two Bumblebee XB3 set-ups, stereo processing for several of the recorded image pairs could not be performed due to poor rectification results with the in-air calibration. These cases would have accordingly lead to significant metric errors; the reported average errors for the in-air calibration are hence a very optimistic, best case estimates.

It can be seen that this quantitative evaluation supports the previous numerical and qualitative observations. The Pinax calibration and rectification leads in all cases to superior results. Most importantly, the errors are significantly smaller than using the state-of-the-art underwater calibration. In addition, Pinax calibration is much more convenient to use as it is based on in-air calibration. The experiments also show that the salinity matters, i.e., if state-of-the-art underwater calibration is done for example in a fresh water pool and the camera is used in the sea, the rectification quality degrades. The Pinax model takes the possible changes of the water salinity into account and is hence not affected by this.

One might expect from a purely theoretical viewpoint that in-air calibration with Pinax and a state-of-the-art underwater calibration that is performed in-situ under the correct water refraction index (underwater with correct WRI) lead to similar, if not identical results. But it is important to note that underwater visibility is worse than visibility in air. This holds even in the best possible cases, i.e., over short distances in a pool. Given the same conditions, i.e., the same number of image frames for calibration, etc., the underwater calibration data is hence noisier due to blur affecting the corner detection. Pinax is hence not only more convenient to use, it also leads to more accurate calibration results in real world applications.

Part III

STEREO SYSTEM DESIGN

Chapter 9

Problem Formulation

9.1 Introduction and Related Work

Cameras have been used in robotics and automation for decades. Some of the most complex tasks such as mobile robot navigation, object recognition or manipulation may require depth information for work [Desouza and Kak, 2002]. In many cases the stereo setup is the preferred choice. However, the choice of the camera, lens and setup parameters (baseline, vergence angle) is rarely discussed. In the recent work [Zhang et al., 2016] some considerations towards choosing the cameras at the stage of system design were presented. However, that work focused only on monocular visual odometry. In [Chen et al., 2007] an attempt was made to formalize the process of designing stereo systems. However, this work focused on optimizing the coverage of the human activities space. On the other hand, different factors must be considered in the case of 3D reconstruction. Estimating the reconstruction errors which stem from the quantization error in a stereo system [Dubbelman and Groen, 2009,Matthies and Shafer, 1987,Herath et al., 2006,Pojar et al., 2012,Zhang and Boult, 2011] requires a considerable amount of work. The quantization error is caused by the fact, that each pixel on the sensor is a patch of surface, not a point, therefore represents a volume in space (see Fig.9.1). This is further discussed in Chapter 10.

This work is mainly motivated by the EU founded project DexROV. A stereo system is needed for various tasks, ranging from general mapping to object recognition and visual servoing. However, the design is limited by the size of the vehicles utilised. Also the expectations regarding modelling accuracy are high, therefore the design process must be studied carefully. In practice there are several factors that may and should be considered at the stage of designing the system. This part of the thesis focuses on the most common design scenarios. All important factors are identified, modelled and finally these formulations are disentangled into a set of constraints. Since the considerations presented here are not application-specific, the proposed solution may be used for designing a stereo system for applications ranging from industrial to research cases, both in normal atmosphere and underwater.

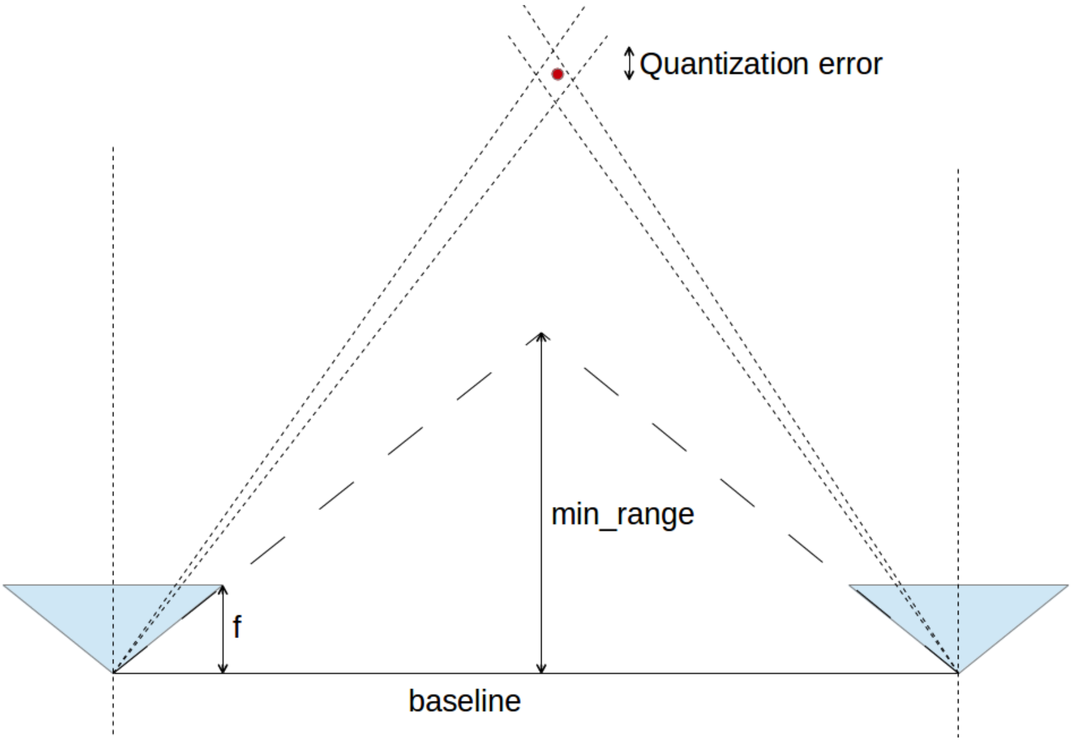


Figure 9.1: Schematic view of the proposed setup and visualization of the quantization error.

9.2 Research Problem

When designing a stereo vision system there are a few decisions which have to be made regardless of applications:

- Selection of the camera,
- Selection of the lens,
- Length of the baseline,
- Vergence of the cameras.

At the same time, these choices are limited by the following constraints:

- Minimum observation distance at which the system should provide 3D coverage,
- Maximum length of the baseline that may be accommodated in the given application,
- Desired accuracy of the 3D reconstruction.
- Desired field of view and area covered by an image at the intended distance to the seafloor.

Below, we discuss each item and define the challenges related to them. There are numerous questions related to the selection of a camera e.g. availability, price etc. When designing a stereo system, a decision has to be made regarding the interface to be used: USB, GigE, Fire Wire etc. Details of each available option and their influence on the system are further discussed in chapter 11. The most important factors to be considered are the resolution of the imaging matrix and the size of a single pixel. Since the same sensor may be installed in various cameras, from this point on we will discuss the selection of a sensor, not the camera. There are some general rules that apply to this selection. A bigger sensor will provide greater field of view at the same focal length of the lens. A bigger pixel size will result in higher sensitivity and better performance in low-light conditions. On the other hand, the bigger the pixel size, the bigger the quantization error. The lens must be compatible with the camera in terms of mount, supported size of sensor, etc. However, the most important factor is the focal length. Increasing the focal length will decrease the quantization error at the cost of decreasing the field of view.

When cameras are selected, they need to be assembled into a stereo rig. The choice of baseline length and vergence is very important for the performance of the whole system. In general, increasing the baseline increases the accuracy of 3D reconstruction by minimizing the quantization error [Rodriguez and Aggarwal, 1990]. On the other hand, it also increases the minimum observation distance. This problem may be solved by verging the cameras, but this in turn narrows the field of view of the system. The baseline is also very often limited by other factors, e.g. maximum length that may be accommodated on an assembly line or a mobile robot. The accuracy that may be expected from the system always decreases as the distance from the camera decreases. However, in an actual setup certain accuracy at a given distance is often required and should be considered when designing such systems. Overall, the relations between the factors to be considered are quite simple, but it is not always obvious how to choose parameters to maximize the performance of a system.

Chapter 10

Quantization Error

10.1 Overview

Quantization error is inherently bound to the way cameras are built and cannot be eliminated entirely. As each pixel on the imaging matrix of the camera has a physical size, it covers a patch of the observed surface, not a single point. The further away from the camera the bigger this patch becomes. When performing 3D reconstruction position of a point can only be known with some limited accuracy, as some differences in its 3D position would not be registered by the camera (Fig.10.1). Usually, quantization error analysis refers to already existing systems. In the case of this analysis, two main approaches are used. The first, landmark based, assumes modelling a specific 3D point, feature, and error on both, left and right image projection [Dubbelman and Groen, 2009, Matthies and Shafer, 1987]. The second approach assumes that whatever is observed by one of the cameras, e.g. the left one, is the reference point. The only source of error comes from bias in the other image [Herath et al., 2006,Pojar et al., 2012]. The way of modelling the error depends on the intended application of the system. When looking for specific markers in the environment the quantization error will occur in both, the left and the right camera. When performing a general reconstruction, e.g. using the sum of absolute differences algorithm for matching features between rectified images, using the second method is more appropriate. In any case, it is important to note that for all reasonable configurations of cameras an error in the reconstructed 3D point is most significant along the Z axis. Quantization error analysis also leads to the conclusion that in the case of the smallest errors, a baseline of infinite length should be used [Rodriguez and Aggarwal, 1990]. In [Zhang and Boult, 2011] an effort was made to define an optimum finite baseline based on extended analysis of the quantization error and minimizing its depth component. However, it only lead to results where the optimum finite baseline can be specified for a given point in space. Therefore, it cannot be used during designing a general purpose stereo vision system.

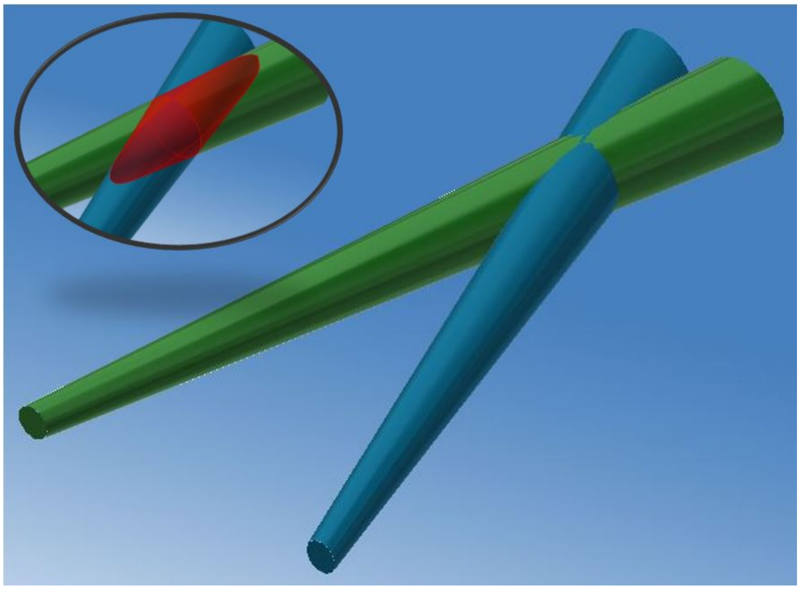


Figure 10.1: Visualization of where the quantization error occurs. Gaussian uncertainty in the image plane increases with the distance. The intersection of two cones created this way, marked red here, is where the triangulated point is expected to be in 3D. The quantization error makes it impossible to differentiate points within this region.

10.2 Modelling Stereo Triangulation Error

Modelling the quantization error, sometimes referred to as the stereo triangulation error, in 3D reconstruction is complex and requires two assumptions to be made. Firstly, it is assumed that an error in the image plane is gaussian. This statement is justified, as the 3D points located in one pixel area are, in general, evenly distributed. Secondly, it is assumed that the 3D volume where the triangulated point is located, can also be modelled using normal distribution. In other words, that the quantization error in 3D is normally distributed. Please note (compare Fig.10.1) that it is true only with some approximation. The true error distribution should have diamond-shaped and not elliptical cross section. However, this approximation is necessary to provide compact and computationally efficient calculations. State of the art methods use linearisation to model the triangulation error [Dubbelman and Groen, 2009, Matthies and Shafer, 1987, Herath et al., 2006, Pojar et al., 2012].

For rectified image pairs the triangulation is conducted as follows. We assume that $\mathbf{l} = [x_l, y_l]$ and $\mathbf{r} = [x_r, y_r]$ are corresponding points in the left and the right image, respectively. Each point has normally distributed error associated with it, described with covariance matrices $\mathbf{V_l}$ and $\mathbf{V_l}$. The goal of triangulation is to find a 3D point $\mathbf{p} = [p_x, p_y, p_z]^T = f_{trian}(\mathbf{l}, \mathbf{r})$ that would be projected to $\mathbf{l}, \mathbf{r}$:

$$p_x = x_l \tfrac{B}{(x_l - x_r)}$$
$$p_y = y_l \tfrac{B}{(x_l - x_r)}$$

$$p_z = f\frac{B}{(x_l - x_r)}$$

where B is the baseline of the stereo system and f is the focal length. For this transformation its Jacobian $\mathbf{J}$ needs to be calculated. Then, f_{trian} may be linearised to calculate the covariance matrix $\mathbf{V_p}$ associated with the 3D point $\mathbf{p}$:

$$\mathbf{V_p} = \mathbf{J}\begin{bmatrix}\mathbf{V_l} & 0 \\ 0 & \mathbf{V_r}\end{bmatrix}\mathbf{J}^T$$

However, it is important to note that triangulation is not linear. Hence, linearisation may introduce an additional error. Furthermore, the non-linearity of the triangulation depends on the angle between the traced rays and camera axis, in other words the non-linearity should be investigated with respect to baseline to depth ratio.

Modelling stereo triangulation error through linearisation is often based on seminal work by [Matthies and Shafer, 1987]. However, eight years after it was published, [Uhlmann, 1995] proposed a new way of calculating how uncertainty propagates through non-linear functions. This method was called unscented transform. It is based on the assumption that it is easier and more accurate to encode the mean and covariance in a set of input points and then reconstruct the probability distribution on the output points. It is therefore essentially a simplified Monte Carlo approach, where many randomly selected points are substituted with a small set of carefully selected points and corresponding weights.

It is worth investigating whether using unscented transform in place of the linearisation will increase the accuracy of quantization error estimation. For this purpose, the following experiment was conducted.

A set of arbitrary setup parameters (intrinsic and extrinsic) and 150 3D points were selected to cover a wide range of baseline to depth ratios. For each of those 3D points, their projections on the left and the right image were found. Image coordinates are assumed to be known with 0.5 pixel accuracy:

$$\mathbf{S} = \begin{bmatrix}\mu_{\mathbf{l}}{}^T, \mu_{\mathbf{r}}{}^T, \mathbf{V_l}(:)^T, \mathbf{V_r}(:)^T \\ . \\ . \\ . \end{bmatrix}$$

where $\mu_{\mathbf{l}}$ and $\mu_{\mathbf{r}}$ are mean values of probability distribution of the left and the right image coordinates, respectively. $\mathbf{V_l}(:)$ and $\mathbf{V_r}(:)$ are vectorised covariance matrices corresponding to those means.

To calculate the ground truth, the Monte Carlo Method is used. For each $\mathbf{s} \in \mathbf{S}$ two sets of 2000 points were randomly selected for the left and the right image, representing given means and covariances. Then, these points were triangulated in an each-to-each manner creating 4 million 3D points. Finally, the mean and covariance were recovered from the triangulated points and saved as a ground truth result.

The first method that will be taken into consideration is linearisation. For each $\mathbf{s} \in \mathbf{S}$ the 3D mean and covariance were calculated accordingly to the method presented earlier in this section.

Finally, a similar procedure has been performed, only this time unscented transform was used. To apply it, samples $\mathbf{s} \in \mathbf{S}$ must be rewritten to a single mean $\mathbf{m_{UT}}$ and a covariance $\mathbf{V_{UT}}$ form for both images:

$$\mathbf{m_{UT}} = \left[\mu_{\mathbf{l}}{}^T \mu_{\mathbf{l}}{}^T\right]^T$$
$$\mathbf{V_{UT}} = \begin{bmatrix} \mathbf{V_l} & 0 \\ 0 & \mathbf{V_r} \end{bmatrix}$$

Then, sample points were selected and 3D mean and covariance were found following the unscented transform procedure [Uhlmann, 1995].

To assess the quality of results of linearisation and unscented transform for each baseline to depth ratio, the results provided by both methods were compared to the Monte Carlo ground truth. For this comparison the Kullback–Leibler divergence was used. The result of this experiment is depicted in Fig.10.2.

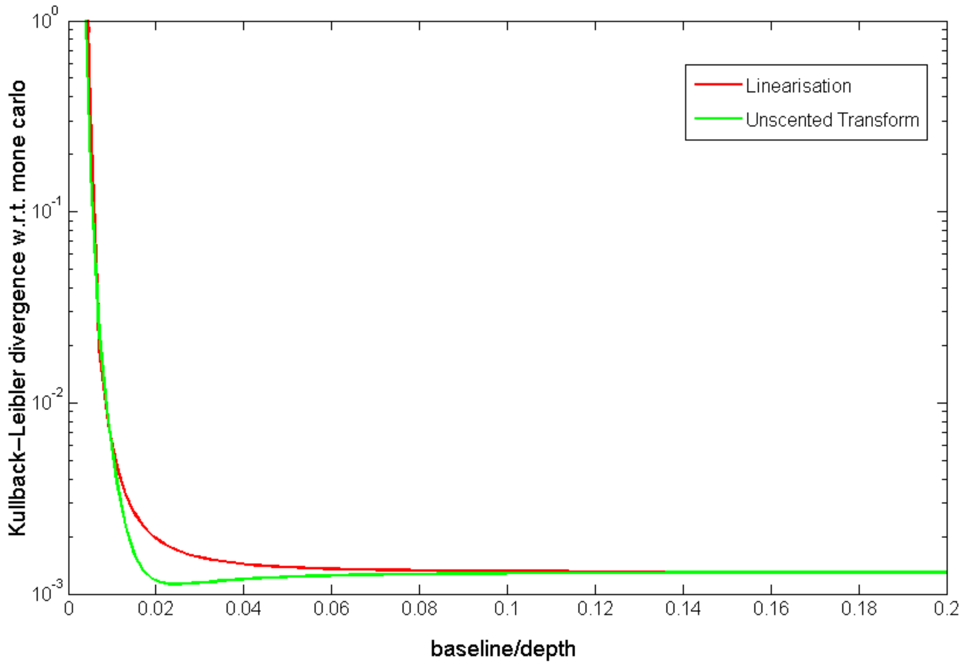


Figure 10.2: Comparison of linearisation and unscented transform used in modelling the stereo reconstruction error.

Using unscented transform resulted in a slight improvement which is especially noticeable far away from the camera. On the other hand, this gain is only visible in baseline to depth ratios in the range from 0.02 to 0.04, i.e. for a stereo system with a 25cm baseline it is 6-12m away from the camera. Stereo cameras are rarely expected to perform well in terms of 3D reconstruction at such long range. However, some applications like feature tracking or navigation may benefit from a more accurate 3D error estimation. That being said, the improvement over linearisation is very modest (please note, that log scale is used on Y axis) and in the overwhelming majority of cases the use of any of these two methods will be sufficient.

Chapter 11

Interface and Sensor

The following chapter will discuss the process on camera interface and sensor type selection. The three most popular options are USB, GigE and FireWire cameras. Since the electrical interface determines the possibility of camera synchronization, this topic is also discussed in other relevant sections. When it comes to sensor type, CMOS and CCD sensors are taken into consideration.

11.1 USB

A USB interface does not offer any support for synchronization on the interface side. In principle, it is possible to achieve this via external camera triggers (e.g. generated by a micro-controller), but the details of the process may differ depending on the camera manufacturer. It may also require the cameras to be equipped with on-board clocks which makes them much more expensive.

The only reasonable choice in terms of bandwidth is USB 3.0. This however requires 9 leads in the connector (three differential pairs, Vcc, Ground, and a signal drain), which is more than is usually found in off-the-shelf underwater cables, creating the need for using custom-made and more expensive cables and connectors. On the other hand, USB has a big advantage of being very popular, which makes integration with different system relatively easy. This should also minimize any compatibility issues in the future.

11.2 Gigabit Ethernet (GigE)

Some GigE cameras have the advantage of supporting the precision time protocol (ptp, IEEE1588). It allows for very precise synchronization between the cameras or synchronization with an external clock, e.g. GPS clock. However, this option makes the cameras quite expensive (up to a factor of two).

It is possible that cameras will need to share the bandwidth with other communication made over the Ethernet, which may significantly limit the available frame rate or make the system unreliable. Furthermore, the GigE protocol does not allow for direct synchronization (when ptp is not supported), also making an external trigger system necessary.

Electrically, it is compatible with off-the-shelf underwater cables (8 wires, with 4 differential pairs), SubConn even markets a line of Gigabit Ethernet cables with and without extra power conductors.

11.3 Fire Wire

Fire Wire has multiple advantages over the other options. Numerous cameras can be "daisy chained", which means they are connected in series without the need for using a switch or a hub. This can dramatically reduce the length and thus the weight of cables required. Additionally, the Fire Wire Camera Sub-Protocol provides interface-level synchronization of all daisy chained cameras. Some camera manufacturers (e.g. Flir / Point Grey) provide software to synchronize cameras across chains. This allows to reuse a proven synchronization method, which significantly eases development. On the other hand, cameras in one chain need to share bandwidth, possibly limiting the maximum frame rate.

Fire Wire 800 (IEEE 1394b-2002) requires 7 leads (two differential pairs, Vcc, Ground, and shield) and is compatible with standard Cat5e twisted pair cables as used in Ethernet connections. This makes it also compatible with standard Ethernet underwater cables and connectors (e.g. by SubConn).

The main drawback is that Fire Wire 800 is not a very common interface any more, making it somewhat harder to find cameras with modern sensors and requiring extension cards (e.g. via PCI Express) on the computer side. Furthermore Fire Wire is very sensitive to cable shielding. This may cause additional problems when working with non-proprietary underwater cables - to the best of our knowledge there are no underwater cables designed for Fire Wire.

11.4 CMOS vs CCD

When choosing a camera, one of the most important components to be considered is its sensor. Sensor type (CMOS/CCD), model and resolution influence numerous important aspects. To begin with, CMOS sensors, due to their design, allow for accessing only selected pixels without reading the whole matrix. In general, CMOS sensors allow for faster reading of the pixel values and therefore provide a higher frame rate. However, high frame rate is not needed for the given task and what is more, it is necessary to keep the field of view as big as possible. On the other hand, CCD sensors are, in general, more sensitive and have lower noise in low-light conditions. Therefore, in underwater applications, it may be the most beneficial to use a camera with a CCD sensor. Any further analyses require the knowledge of the sensor's resolution and pixel size: this depends on the sensor model. A detailed description of the selection method is presented in the following chapter.

Chapter 12

Optimization of the Setup Parameters

12.1 Redefining the Minimum Range of the System

The cameras are assumed to be identical, with the same lenses and they are verged symmetrically. The setup will be analysed in the XZ plane. This assumption allows for simpler the analysis of all important factors without any losses. Presented below is the notation used in the following analysis (compare Fig.12.1):

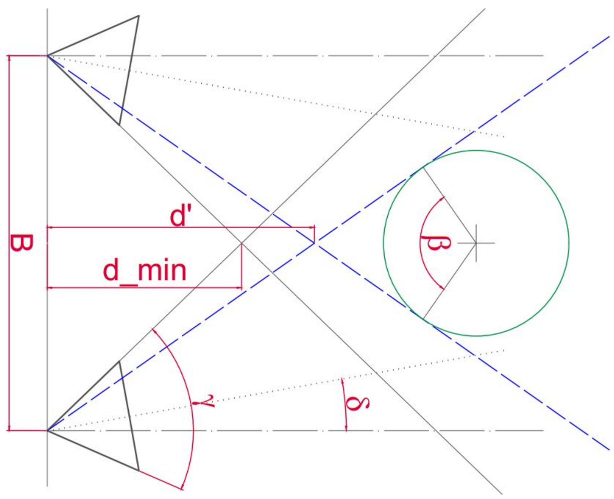


Figure 12.1: Overview of the considered situation and the notation.

- B is the baseline lengh
- f is the focal length of the lens
- γ is the field of view of the camera

Figure 12.2: Example of influence of the radius of the object on stereo imaging. Three cameras are 25cm from each other (50cm between the left and the right one). The cylinders are 120mm, 80mm and 45mm in diameter.

- δ is the vergence angle
- $s.w$ is the width of the sensor s in pixels
- $s.p$ is the size of a pixel of the sensor s
- d_{min} is the theoretical minimum range of the system
- β is the angle defining minimum arc of the object that should be observable
- d' is the minimum range corresponding to the β angle

Usually, the minimum range of the system is defined by the border pixels of the two cameras (Fig. 9.1). However, this is only valid when observing a planar surface. To generalize this case, we will approximate the observed non-zero curvature of the object with a sphere or a cylinder (a circle in a projection on the XZ plane). When observing an object like this, we are interested in reconstructing its surface. It is only possible for an arc that can be seen by both cameras. Let's define this arc with an angle β. This in turn will redefine the minimum range as d' (Fig.12.1). Selection of β will depend on the size of the objects that are expected to be modelled and the task given to the system. In a limit case, for a plane, the β may be infinitely small. To visualize this concept, compare Fig.12.2. Three cameras were positioned every 25 cm. In front of them, three cylinders were placed (120mm, 80mm and 45mm in diameter). It is visible that as the diameter gets smaller, the size of the surface patch that may be reconstructed also shrinks, but the angular measure of this patch remains constant. It must be noted that d' defined this way is shorter than the actual distance to the surface of the object. However, it was decided to leave it this way, as it will make the results of the algorithm more robust when modelling objects with more complex shapes, only approximated with a cylinder/sphere.

12.2 Non Verged Case

In this section the simplest case will be analysed. We assume that no vergence is allowed ($\delta = 0$). This allows for maintaining the field of view of the system identical to the field of view of the cameras used. Therefore, it is also the most versatile and commonly used in e.g. mobile robotics. Below, we formulate the constraints resulting from the input data.

12.2.1 Maximum baseline constraint

There are two factors limiting the maximum baseline that may be used. The first one is the maximum length allowed by the user B_{usr}, the second stems from the minimum range / β constraint. It must be also noted that if the following analysis is to be object size invariant, the minimum β_{min} angle must be specified. Its value is defined with an assumption that the whole object needs to be in the field of view, thus the shortest arc can be observed when the two light rays, tangential to the cylinder/sphere, correspond to pixels on the borders of images (Fig. 12.1).

$$\beta_{min} = 2 * (90 - \frac{\gamma}{2}) \tag{12.1}$$

If the user inputs $\beta < \beta_{min}$, this value is increased to the value of β_{min}. As the minimum distance d' does not denote the distance to the object's surface, this preliminary analysis can be performed without specifying the size of the object.

Having β specified, the maximum baseline can be calculated:

$$B' = \frac{2 * d'}{\tan \frac{\beta}{2}} \tag{12.2}$$

Finally, the first constraint is formulated

$$B_{max} = min(B', B_{usr}) \tag{12.3}$$

12.2.2 Minimum baseline constraint

The minimum baseline constraint arises from the minimum accuracy specified by the user. As discussed earlier, even if we assume calibration and matching to be perfect, the quantization error remains. The user is expected to specify a set of distances d^i and accuracies a^i expected at these locations. Let's consider a point P lying on a centre line between the cameras at the distance of d^i. To model the quantization error we assume that the x coordinate of the projected points can vary by $\frac{s.p}{2}$. This constraint is specified iteratively. The baseline is increased by a fixed step and the accuracy is tested if it already fulfils the assumed value. This way a minimum baseline B^i_{min} is specified (one for each given distance/accuracy pair). The final constraint is:

$$B_{min} = max(B^1_{min}, ..., B^i_{min}) \tag{12.4}$$

This way, all the specified accuracy requirements will be fulfilled.

12.2.3 Full algorithm

Once both the minimum and the maximum baseline constraints are calculated, the solution may be provided. Three major cases may be specified. In the first case, $B_{min} > B_{usr}$, there is no solution. It means that even at the maximum baseline allowed by the user it is impossible to achieve the required accuracy. In the second case, $B_{max} < B_{min} < B_{usr}$. This means that there is no solution when using two cameras, but adding more cameras may potentially help. A pair with a wider baseline will guarantee the required accuracy further away and a shorter baseline will provide the coverage of the objects close to the system. Of course in some cases this may not be an option, as very high accuracy at a short distance may be required. Finally, when $B_{min} < B_{max}$, a solution may be given. In general, any baseline length within this range will satisfy the requirements specified by the user. However, it needs to be noted that additional sources of errors may occur, both in the quality and density of the reconstruction, so selecting the border values of the returned range is not recommended. The algorithm requires the following input:β, d', B_{usr}, a list of sensors to be considered S, a list of available focal lengths F and a list of required accuracies at given distances (d^i, a^i). $NULL$ in Algorithm 3 encodes the case with no solution.

12.3 Limited Vergence Case

The second case considered describes a situation when the user allows for vergence, but wants to maintain the unlimited maximum range. In other words $\delta < \frac{\gamma}{2}$.

The equation for β_{min} (12.1) must be modified:

$$\beta_{min} = 2 * (90 - \gamma) \tag{12.5}$$

A vergence angle δ must also be provided. If the solution may be provided, δ is calculated as:

$$\delta = 90 - \arctan(\frac{2 * d'}{B}) - \frac{\gamma}{2} \tag{12.6}$$

The changes in the Algorithm3 must also be made accordingly to (12.6). The vergence angle δ is also limited to positive values – there is no point in turning the cameras away from each other. The results of δ calculation are given for B_{min} and B_{max}.

12.4 Unlimited Vergence Case

In the last analysed scenario, vergence δ is not limited. This may be useful e.g. when using lenses with a long focal length to model an enclosed space of known relative

Algorithm 3: Stereo system design algorithm, non verged case.

for *each* $s \in S$ **do**
  for *each* $f \in F$ **do**
    $f_p = \frac{f}{s.p}$
    $l = [-s.w; 0; f_p], r = [s.w; 0; f_p]$
    $l.normalize(), r.normalize()$
    $\gamma = \arccos(l^T * r)$
    if $90 - \frac{\gamma}{2} > \frac{\beta}{2}$ **then**
      $\beta = 2 * (90 - \frac{\gamma}{2})$
    $B_{max} = \frac{2*d'}{\tan(\frac{\beta}{2})}$
    for *each* (d^i, a^i) **do**
      $B = 0.02$
      $step = 0.005$
      while *not_solved* **do**
        $B+ = step$
        $P = [-\frac{B}{2}; 0; d^i]$
        $Img = \frac{P*f}{P.z}$
        $Img_{min} = Img_{max} = Img$
        $Img_{min}.x = Img.x + \frac{s.p}{2}$
        $Img_{max}.x = Img.x - \frac{s.p}{2}$
        $P' = \frac{Img_{min}}{Img_{min}.x} * \frac{B}{2}$
        $P'' = \frac{Img_{max}}{Img_{max}.x} * \frac{B}{2}$
        $e = |P'.z - P".z|/2$
        if $e < a^i$ **then**
          solved
          $B^i_{min} = B$
        else if $B < B_{usr}$ **then**
          solved
          $B^i_{min} = NULL$
    if $max(B^1_{min}, ..., B^i_{min}) == NULL$ **then**
      $Results(s, f) = NULL$
    else if $max(B^1_{min}, ..., B^i_{min}) > B_{max}$
    $\&\&min(B^1_{min}, ..., B^i_{min}) > B_{max}$ **then**
      $Results(s, f) = NULL$
    else if $max(B^1_{min}, ..., B^i_{min}) > B_{max}$
    $\&\&min(B^1_{min}, ..., B^i_{min}) < B_{max}$ **then**
      $Results(s, f) = MORE\_CAMERAS\_NEEDED$
    else
      $Results(s, f) =< B_{min}, B_{max} >$

output: *Results*

position with high accuracy. In this case, there is a risk that one of the distances for which the desired accuracy was specified, will be beyond the maximum range d_{max} of the system. Therefore, a new constraint for B_{max} must be formulated.

$$\gamma < \arctan(\frac{2 * d_{max}}{B_{max}}) - \arctan(\frac{2 * d'}{B_{max}}) \tag{12.7}$$

However, the B_{max} value cannot be extracted from (12.7). Therefore, the Algorithm 3 is modified by adding Algorithm 4 to solve this constraint iteratively. A simple solver with fixed step is proposed here, as this is only used for specifying initial clues for the further design decisions and an exact solution is not necessary.

Algorithm 4: Additional constraint for B_{max}.

$\alpha = \arctan(\frac{2*d_{max}}{B_{max}}) - \arctan(\frac{2*d'}{B_{max}})$
while $\alpha > \gamma$ **do**
  $B_{max} -= 0.05$
  $\gamma < \arctan(\frac{2*d_{max}}{B_{max}}) - \arctan(\frac{2*d'}{B_{max}})$

12.5 Analysing the Chosen Setup

The algorithm presented earlier in this chapter provides the range of baselines for which the given conditions are be fulfilled. Choosing a baseline closer to the maximum value increases the accuracy, while choosing a shorter baseline increases the observable part of the surface (β). Both values are subject to additional sources of errors. The predicted accuracy may be lowered by errors in camera calibration, motion blur, bad light conditions, etc. The β angle may decrease due to the above factors, but also due to a featureless texture and a quantization error. In this case, the quantization error will manifest on the side of the cylinder where within one pixel both the background and a part of the cylinder are visible. Some of these effects depend on the size of the object. Therefore, a second step of the design process is introduced. After selecting the sensor, lens, baseline, etc. all these values are analysed to calculate the predicted parameters of reconstruction accuracy and β angle. In this instance, β is calculated together with the predicted influence of quantization error. Since this influence depends on the cylinder's diameter, it is presented in the form of a graph. The predicted value of β corrected for quantization error (denoted as β_c) can be calculated as:

$$\sigma = \arcsin(\cos(\alpha) - \cos(\alpha)\tan\alpha\tan(\frac{\beta}{2}) - \frac{B\tan\alpha\cos\alpha}{2R\cos(\frac{\beta}{2})}) \tag{12.8}$$

$$\beta_c = \beta - 2(90 - \alpha - \sigma) \tag{12.9}$$

where α is the angle defined by the size of a single pixel for the given focal length f and σ is the auxiliary angle used in calculations.

The accuracy of the system is calculated by estimating the maximum quantization error as in Algorithm 3. An example use of this method, applied in the DexROV project, is presented and discussed in Chapter 15, Part V of this thesis.

12.6 Evaluation

To evaluate the quality of the analysis proposed in Sec. 12.5, the following experiment was conducted. Three different systems were assembled. One with a baseline $B = 0.25m$, another with $B = 0.1m$ and a final system with a baseline of $B = 0.4m$. The system with $B = 0.25m$ is considered the reference setup, while longer and shorter baselines are tested to evaluate the influence of setup parameters on the reconstruction capabilities. For each system an analysis was performed, as described in Sec. 12.5. Then, the reconstruction capabilities were verified empirically. A cylinder was placed in front of the system (compare Fig. 12.3), the lines printed on its surface were applied every 10 degrees to simplify the measurement of the β angle. This way, the predicted values can be compared to the real images. On the other hand, it is important to note that this is only a theoretical value, whereas the actual reconstruction capabilities depend on many other factors, i.e. texture, light conditions, reconstruction method etc. The results of this comparison are presented in Table 12.1

Table 12.1: Predicted values of β_p and measured β_m for different systems.

Baseline / Distance	0.1m	0.25m	0.4m
0.3m	$\beta_p = 143$ $\beta_m = 135$	$\beta_p = 120$ $\beta_m = 115$	$\beta_p = 98$ $\beta_m = 95$
0.5m	$\beta_p = 146$ $\beta_m = 140$	$\beta_p = 133$ $\beta_m = 125$	$\beta_p = 118$ $\beta_m = 115$

To minimize the errors coming from the aforementioned sources of reconstruction errors, the β angle of the arc visible on both cameras is read manually with the help of lines printed on the cylinder. This measurement has an accuracy of up to 5 degrees. Just as expected, the shorter the baseline, the bigger part of the object is seen by both cameras at the same time. Also, the measured values are only slightly smaller than predicted, so this part of the design process worked as intended.

In the second experiment, the accuracy prediction for the system was evaluated. The same three stereo systems were used: with $0.1m$, $0.25m$ and $0.4m$ baseline. Each system recorded the calibration pattern at $0.5m$, $1m$ and $2m$ distances. A calibration pattern was used, as the markers on it can be detected with subpixel

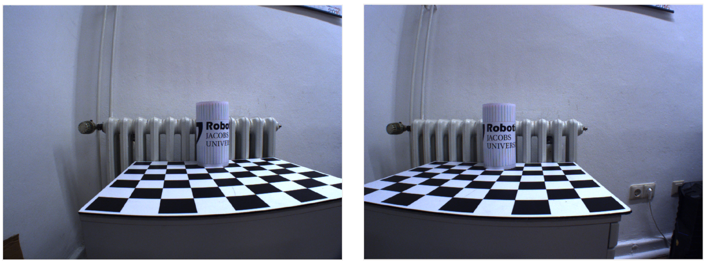


Figure 12.3: Example view of the cylinder as seen by the left and the right camera, used in evaluation process.

accuracy, providing ground truth reference. The positions of these markers were later rounded to full pixels and after triangulation, their 3D positions were compared to the ground truth. This simulates the effect of quantization error when reconstructing the position of a feature in the environment. Please note that when performing a full 3D reconstruction, a different approach to modelling errors should be used and only the quantization error of one of the cameras should be taken into account. In other words, it is assumed that the patch of the surface seen by a pixel of one of the cameras is the reference feature and the error is only the result of matching this patch by the second camera (compare Chapter 10 and [Herath et al., 2006, Pojar et al., 2012]). However, the choice of a modelling method does not influence the algorithm, so validation is performed only once. The results of this experiment are presented in Fig. 12.4. As the algorithm predicts the maximum error resulting from quantization, it is also compared to the maximum error recorded in the experiment.

The obtained results have confirmed that the measured values align with the predicted ones quite well. As the distance to the object increases and the maximum reconstruction error gets very large, the measured error remains a bit smaller. This is simply due to the fact that the worst possible scenario did not occur, yet the prediction worked as intended, providing a clue on what the worst case scenario may be.

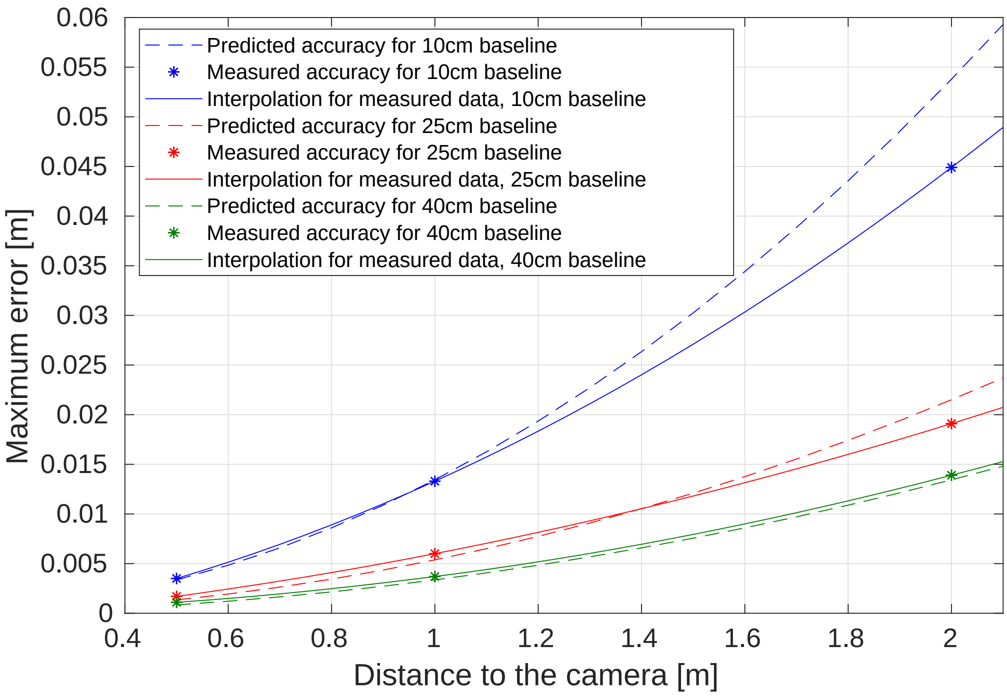


Figure 12.4: Evaluation of the accuracy prediction algorithm.

Part IV

IMAGE ENHANCEMENT

Chapter 13

Underwater Imaging

13.1 Overview

Underwater imaging suffers from exceptionally bad light conditions and significant refraction-based distortions. Image deformations have already been discussed in Part II. However problems related to underwater light propagation are much more complex. Light is strongly attenuated and scattered. Basis of underwater image formation model was formulated seminal paper by [Jaffe, 1990]. Further analysis of the image formation process can be found in multiple subsequent papers, e.g., [Schechner and Karpel, 2005, Chiang and Chen, 2012, Yamashita et al., 2007]. Many attempts have been made to improve the quality of underwater images. An overview of these results is given for example in [Schettini and Corchs, 2010]. As identified in [Schechner and Karpel, 2005] and [Treibitz and Schechner, 2009] backscattering is the dominant degradation component. A physically accurate, polarization based method for haze removal is proposed in [Schechner and Karpel, 2005]. However this method requires artificial light with a polariser and to take two images from the same viewpoint with the analyser being physically differently oriented. Therefore this method is difficult, or in some cases impossible, to apply, especially on moving vehicles. On the other hand there is a well known method for haze removal from images taken in air, called Dark Channel Prior (DCP) [He et al., 2011]. Some attempts to adapt this method for underwater conditions were made [Jr et al., 2013, Cheng et al., 2015]. Details of this research is discussed later.

Work presented in this Part is motivated by the DexROV project. These results have been published earlier in the proceedings of the IEEE Oceans'17 conference, titled *"Underwater image haze removal with an underwater-ready dark channel prior"* ([Łuczyński and Birk, 2017]).

From the perspective of the project, when using artificial lightning backscattering effects may be especially visible. On the other hand, 3D stereo reconstruction is to be used, therefore haze removal, even at a cost of degrading the colour information, is crucial (compare Fig.13.1). As our initial tests showed, state of the art haze removal methods did not lead to sufficient results for robust stereo processing - they either cannot be applied to a moving ROV ([Schechner and Karpel, 2005, Schechner and Averbuch, 2007]) or the improvement was considered to be insufficient ([Jr et al., 2013, Cheng et al., 2015]). The goal of the research presented in this Part is to modify the DCP so that it can be applied to underwater images. The proposed method is be compared to state of the art methods based on DCP, namely [Jr et al., 2013, Cheng et al., 2015], and the physically more accurate polarization based method [Schechner and Karpel, 2005, Schechner and Averbuch, 2007].

Figure 13.1: ROV with the DexROV system attached. Please note that significant amount of light is scattered around the cameras.

13.2 Image Formation Model

An underwater image formation model is described in multiple papers [Schechner and Karpel, 2005, Jaffe, 1990]. For our work the following model is used. The signal forming the image is a sum of three components:

$$E_T = E_d + E_f + E_b \tag{13.1}$$

13.2.1 Direct Transmission and Light Attenuation

E_T is a total radiance sensed by the camera. The first component forming E_T is the direct transmission E_d of the radiance reflected from the object E_o. Light, after being reflected from the scene, is attenuated. Attenuation is responsible for absorbing light with the distance from the source. This absorption depends on the wavelength, hence causing change in perceived colour of the observed scene:

$$E_d = E_o e^{-c_\lambda r} \tag{13.2}$$

Where r is a distance and c is a total attenuation coefficient. Subscript denotes that c depends on the wavelength.

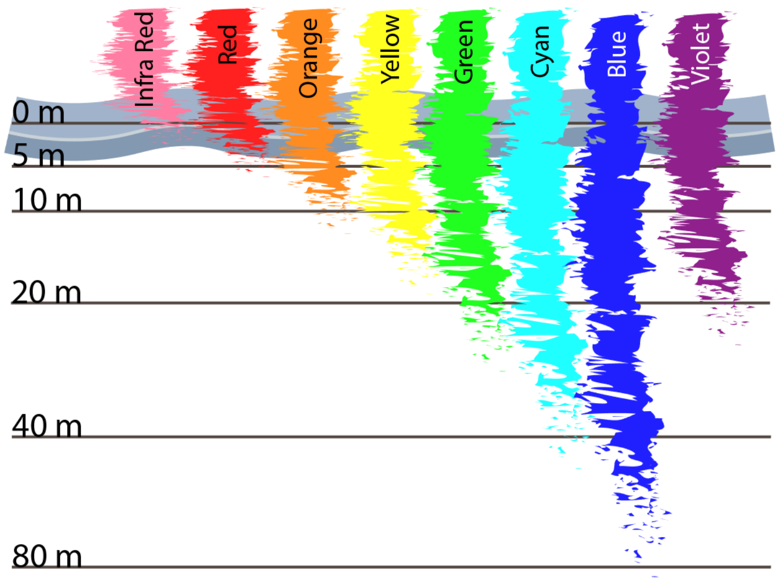


Figure 13.2: Attenuation of light underwater.

13.2.2 Forward- and Backscattering

The two remaining components of E_T are caused by scattering and can be further divided into two phenomena: forward E_f and back scattering E_b. The underlying physics for both is the same but the source of the scattered signal differs, they therefore have different influences on the image. Forward scattering occurs when light reflected from the scene scatters on its way to the camera. This results in slight blur in the image that can be described by convolution:

$$E_f = E_d * g_r \tag{13.3}$$

Where g_r is a point spread function (PSF) and is parametrized with distance r. There are several models of the underwater PSF [Voss, 1991], but their details do not matter to the work presented here and they will hence not be discussed any further.

Finally, backscattering, sometimes called the veiling light, occurs when an ambient light is being scattered and reflected back to the camera, adding a new signal

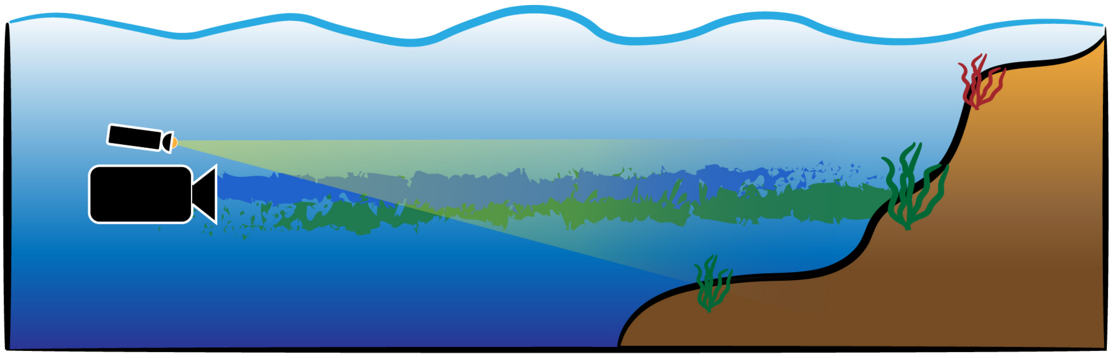


Figure 13.3: Underwater image formation scheme: green denotes light reflected from the object. This signal is getting weaker with distance, as the light is attenuated and scattered. Blue shows the veiling light caused by the artificial source and natural light coming from the surface. It is scattered and partially reflected back to the camera. The backscatter signal is getting stronger with the distance and may, eventually, dominate the direct transmission and occlude the observed scene. The goal of the haze removal is to extract and reduce the backscattered signal (blue), so that even if the remaining direct signal (green) is weak it is possible to retrieve the information.

component to the registered image. The further the scene from the camera, the stronger and more visible the backscattering component gets (compare Fig. 13.3). It can be compared to observing the environment in the fog. When no natural light is present the amount of backscattered signal depends on the overlap between the line of sight and the cone of artificial light being used. For that reason it may be desirable to use light separated from the camera to illuminate the scene from the side. However in most practical applications, e.g., when using ROV, the light is placed close to the camera, thus creating relatively strong backscatter. Backscatter signal caused by ambient light may be calculated as:

$$E_b = B_{\mathrm{inf}}(1 - e^{-c_\lambda r}) \tag{13.4}$$

Where B_{inf} is a background signal, i.e., what would be seen if the camera was looking into open water with no objects in front of it. It is important to note, that it depends on light conditions, therefore may change, especially when only natural light is present. Furthermore, the bigger the distance of the camera to the observed scene, the stronger the veiling light. Because of that, the image cannot be corrected as a whole - for each pixel a different distance may apply, therefore different correction operations per pixel are necessary.

13.2.3 Practical Notes

From the practical perspective, when 3D stereo reconstruction is the main goal, attenuation, forward and back scattering have different influence on the sensed image. Attenuation changes the colour and lowers the overall intensity but does not influence the stereo 3D reconstruction significantly. Forward scattering introduces a slight blur, effectively smoothing the image, which may be beneficial for matching correspondences between the images. Finally back scattering was identified [Schechner and Karpel, 2005] as the main source of the image degradation and reducing this phenomenon is therefore most important for accurate 3D reconstruction.

Chapter 14

Dark Channel Prior in Underwater Conditions

14.1 Dark Channel Prior

The mechanism causing veiling light underwater is also present in air, only on a much smaller scale. It is caused by water and dust particles and may be observed on larger distances. In recent years, the Dark Channel Prior (DCP) [He et al., 2011] became a popular method in this context. The dark channel I^{dark} of a given image I is defined as:

$$I^{dark}(x) = \min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} I^c(x) \right) \tag{14.1}$$

The dark channel is formed by taking a minimum value from all the colour channels within a patch around the given pixel. The Dark Channel Prior is based on the observation that in most of the non-sky patches (in outdoor images) at least one colour channel has some pixels with very low intensity. Therefore:

$$I^{dark} \to 0 \tag{14.2}$$

For the sake of a more compact notation the double min operator used to calculate the dark channel will be further denoted as:

$$\min_{y \in \Omega(x)} \left(\min_{c \in \{r,g,b\}} ... \right) = DC(...) \tag{14.3}$$

As shown in [He et al., 2011] usually bright patches in the dark channel on non-sky regions appear due to the backscattering. Therefore it can be used to estimate and correct its influence on the image. Of course this method has its limitations, e.g., it cannot be used to process a snowy scene or a view at big white wall from close distance. However, in air the need for correcting backscattering occurs only when imaging outdoor scenes at long ranges. Therefore, DCP can be successfully used in most cases. The haze removal in (long range) outdoor scenes with DCP assumes a similar model as in underwater vision:

$$J(x) = I(x)t(x) + A(1 - t(x)) \tag{14.4}$$

where J is an image without the haze, A is a global atmospheric light - which corresponds to B_{inf} - and $t(x)$ is a transmission function:

$$t(x) = e^{-\beta_\lambda r} \tag{14.5}$$

β corresponds to the underwater attenuation coefficient c. In the first step A is estimated. It is based on an other assumption, namely that atmospheric light is white and is calculated by taking the most haze-opaque region of the image (which can be found as the brightest region of the dark channel image). Then both sides of (14.4) are normalized with A:

$$\frac{J(x)}{A} = t(x)\frac{I(x)}{A} + 1 - t(x) \tag{14.6}$$

Then, the dark channel is calculated on both sides:

$$DC\left(\frac{J(x)}{A}\right) = t(x)DC\left(\frac{I(x)}{A}\right) + 1 - t(x) \tag{14.7}$$

Using the dark channel prior on the haze-free image in the equation:

$$DC\left(\frac{I(x)}{A}\right) = 0 \tag{14.8}$$

This finally leads to the estimation of the transmission function:

$$t(x) = 1 - DC\left(\frac{J(x)}{A}\right) \tag{14.9}$$

In the last step, the estimation of the transmission function is refined with soft matting [Levin et al., 2008]. This step improves the quality of the corrected image, but also takes quite a long time to compute making the online application of this method impossible.

As noticed earlier, the haze model underlying the dark channel prior correction method corresponds to the underwater backscatter model. Therefore it is immediately obvious that there is a significant potential in this method for underwater applications where the haze in the images is much more pronounced. Unfortunately, the dark channel prior cannot be used directly on underwater images. The method assumes that the backscatter component is white, which is true in air, but the heavy attenuation underwater causes the red wavelength to disappear very quickly (compare Fig. 13.2), leaving a (distance dependent) blueish hue of the veiling light. This also causes the dark channel to be completely black. If brighter regions appear, they usually correspond to the light patches of, e.g., sediment close to the camera (Fig.14.1).

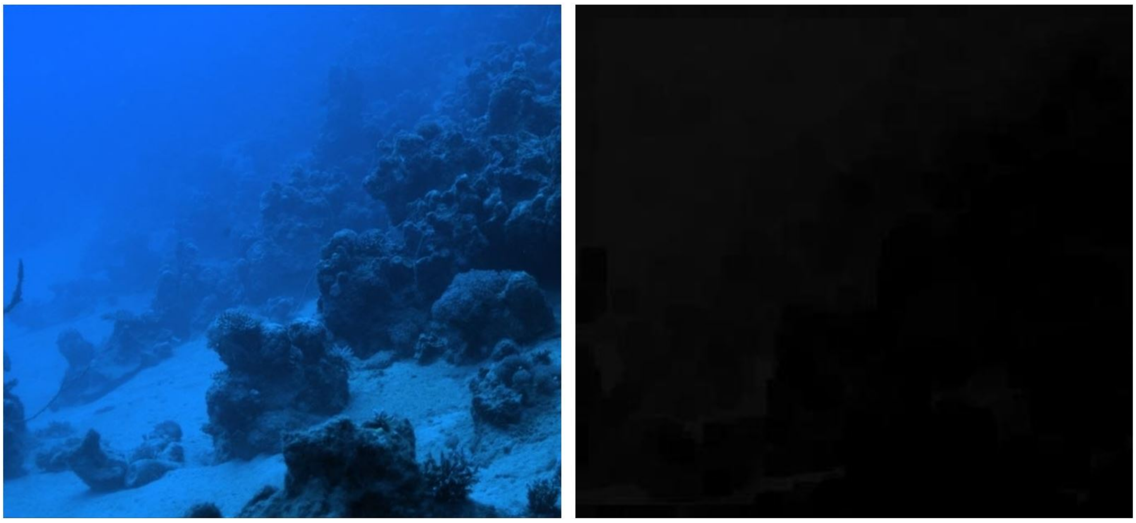


Figure 14.1: A raw underwater image (left) and its dark channel (right). Due to the high attenuation of the red light, the dark channel is almost completely black. The image is taken from [Schechner and Karpel, 2005, Schechner and Averbuch, 2007]; the image is shared for non-commercial use: http://webee.technion.ac.il/ ∼ yoav/research/ underwater.html. It is one of the images used here for comparing our results to other DCP-based methods and the physically accurate approach presented in [Schechner and Karpel, 2005, Schechner and Averbuch, 2007].

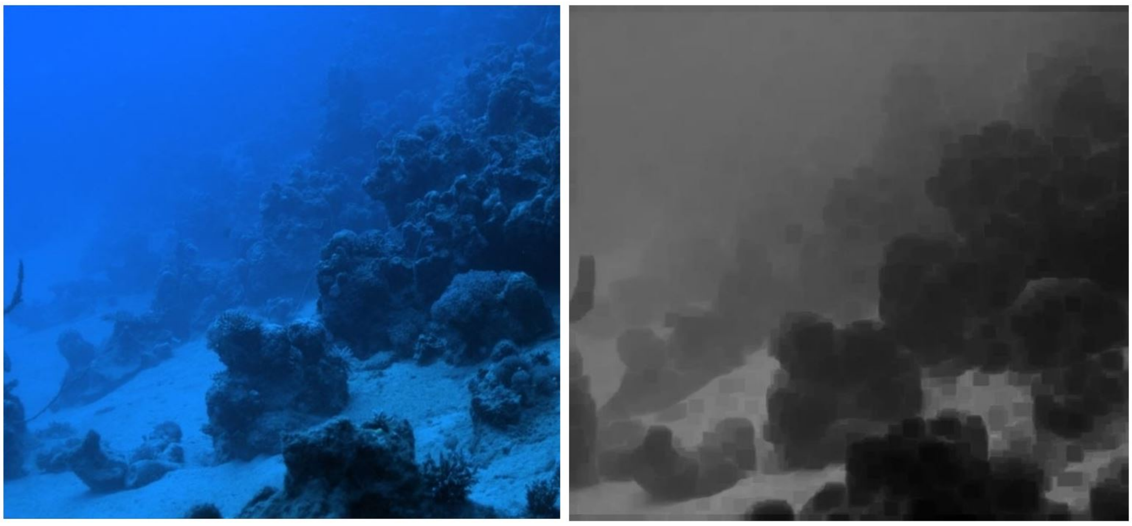


Figure 14.2: Raw underwater image (left) and its underwater dark channel calculated with [Jr et al., 2013] (right). There is a visible improvement compared to the classic DCP, but the result is still not ideal - the heavily hazed part of the image is not recognized as such.

14.2 Underwater Adaptations to Dark Channel Prior

Two major attempts to adjust the dark channel prior to underwater conditions were made before. In [Jr et al., 2013] it was noticed that attenuation of the red light is very strong. Therefore it was proposed to use green and blue channels only. The authors called this method Underwater Dark Channel Prior (UDCP). This modifies (14.3):

$$\min_{y \in \Omega(x)} \left(\min_{c \in \{g,b\}} ... \right) = UDC(...) \tag{14.10}$$

Results, when using UDCP, are a bit better than those with DCP but the calculated dark channel is still not correct (Fig. 14.2). The brightest region in the dark channel is the patch of sand close to the camera. This is understandable, as DCP in general does not handle regions like that well. A more severe drawback is that the heavily hazed background of the image is relatively dark, where in the optimal case it should be white.

The authors of [Cheng et al., 2015] build on the observation that the red light is attenuated so strongly that its presence implies a weak backscattering signal. Therefore, the complement of the red channel is used when calculating the dark channel. This method is here referred to as Red-Dark Channel Prior (RDCP). Equation (14.3) is therefore modified to:

$$\min_{y \in \Omega(x)} \left(\min_{c \in \{1-r,g,b\}} ... \right) = RDC(...) \tag{14.11}$$

Similar to UDCP, this method shows some improvements over DCP, but the dark channel calculated with this method is far from optimal (Fig. 14.3).

For example results of haze removal for all three methods please refer to Fig. 14.4.

14.3 Proposed Method

Observations made by authors of [Jr et al., 2013] and [Cheng et al., 2015] are indisputably correct. However, we believe that this problem should be handled differently. The observation underlying the original DCP can be formulated as follows: at least one colour channel has some pixels with very low intensity for most of the non-sky patches. This can be reformulated to: the stronger the backscattering component, the whiter the region gets. Formulating the problem like this gives a better intuition for the adjustment of DCP to underwater: the backscattered light is (predominantly) blue in the underwater scenario, not white. Furthermore, it is only important for the proper estimation of the global atmospheric light A. If that step is performed correctly, the normalization step (14.6) removes the influence of the colour light.

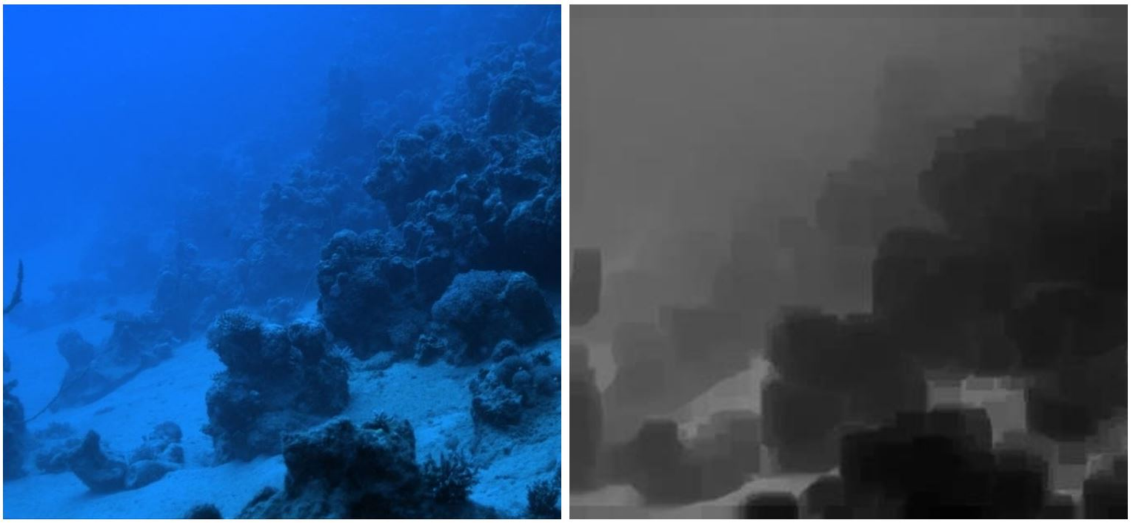


Figure 14.3: Raw underwater image (left) and its underwater dark channel calculated with [Cheng et al., 2015] (right). There is a visible improvement comparing to the classic DCP, but the result is similar to the UDCP method and still not correct - the heavily hazed part of the image on is not recognized as such.

Figure 14.4: Comparison of the literature dark channel methods. From the left: raw image, classic DCP [He et al., 2011], UDCP [Jr et al., 2013] and RDCP [Cheng et al., 2015]

Therefore, the core of DCP does not need to be modified, but the method of estimating A needs to be changed instead. Before estimating the global atmospheric light, all the colours in the image should be shifted in the colour space so that blue becomes white. After this modification A may be estimated from the dark channel of the modified image and the rest of the procedure remains as in [He et al., 2011]. The modified procedure is as follows. To transform blue into white, the RGB coordinate system is shifted (compare Fig. 14.5):

$$R' = 255 - R \tag{14.12}$$

$$G' = 255 - G \tag{14.13}$$

$$B' = B \tag{14.14}$$

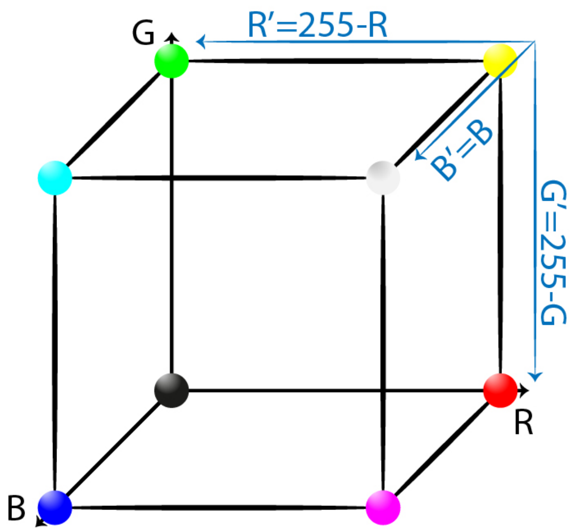


Figure 14.5: Proposed colour transformation for easy and accurate estimation of B_{inf}.

For the image with the shifted RGB coordinate system, its dark channel is calculated. As visible in Fig. 14.7, the calculated dark channel represents the veiling light in the image much better than the other methods (compare Fig. 14.2 and 14.3). At this point, A may be estimated by taking the colour corresponding to the brightest value in the dark channel. After that, the original image is normalized and processed without any modifications to the original method.

14.4 Results

14.4.1 Comparison to Other Dark Channel Based Methods

In a first test, our method is compared to RDCP and UDCP. A small set of sample images is processed with all three methods for direct comparison. The results are

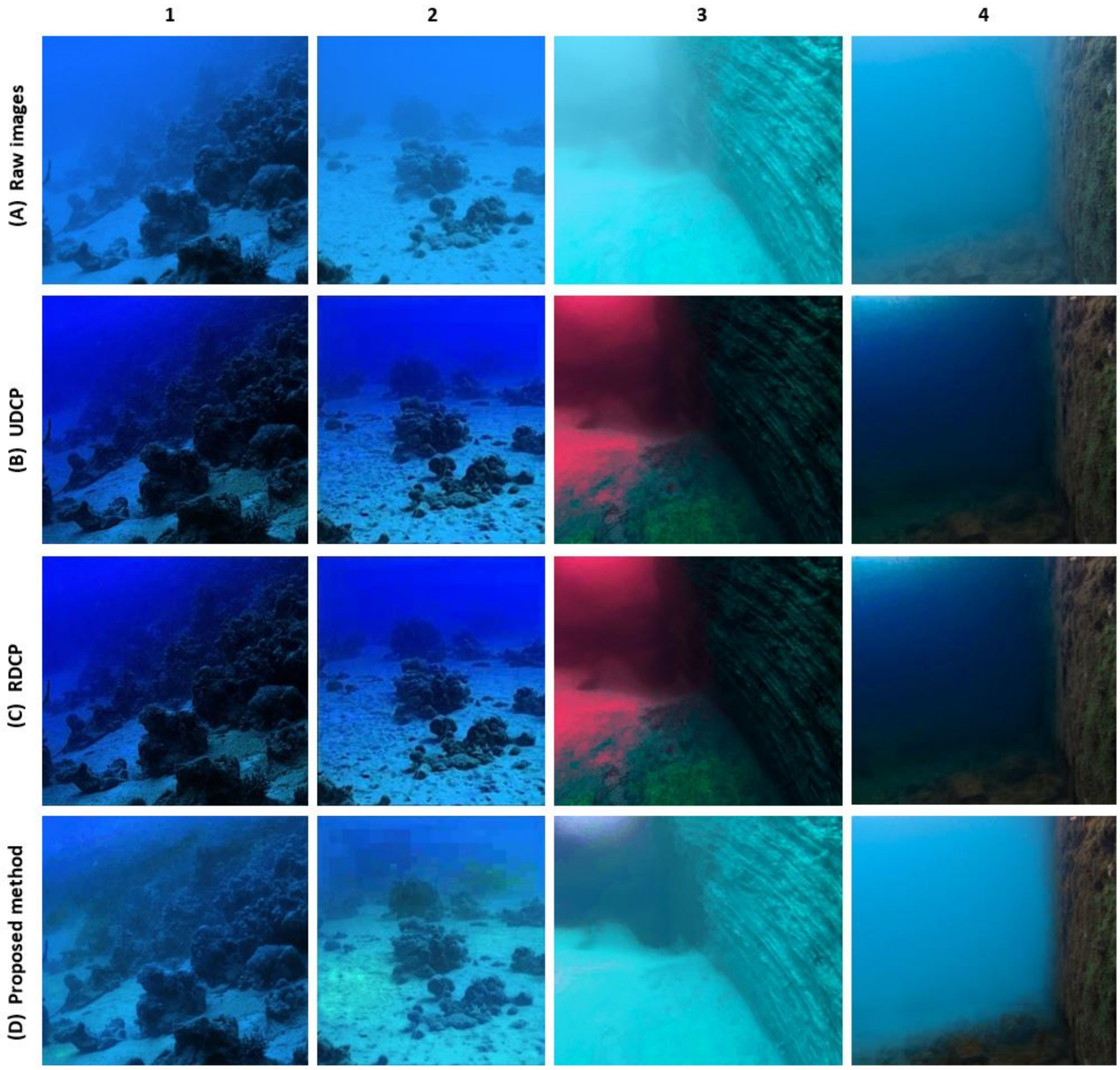


Figure 14.6: Comparison on different example images. Image 1 and 2 are taken from [Schechner and Karpel, 2005, Schechner and Averbuch, 2007], shared by authors for non-commercial use: http://webee.technion.ac.il/ ∼ yoav/research/underwater.html. Images 3 and 4 were recorded during MORPH project trials on the Azores. Following rows present results of image correction with different methods.

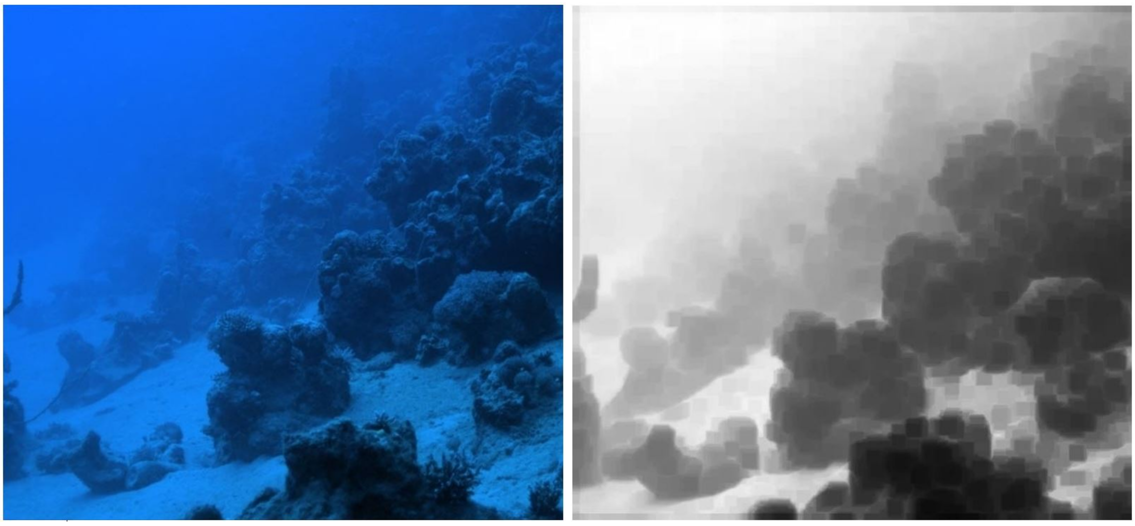


Figure 14.7: The raw image and the corresponding dark channel calculated with proposed method. Please note that the heavily hazed region in the top left corner of the image is white in the dark channel, correctly indicating very strong presence of the backscattering component.

presented on Fig. 14.6.

Depending on the contrast of the raw image, the recovery results may vary. In the worst case, the gain in quality is minimal or none, but the colours after proposed correction are never distorted, as it happened for RDCP and UDCP, e.g., with sample image 3. Furthermore, there is almost no loss in brightness. Overall the results are promising, as they outperform competing dark channel based methods. The adaptation proposed in this paper adapts the DCP to underwater conditions, preserving all the advantages of this method. On the other hand all the drawbacks of the original work are also present here: computation takes a lot of time and, for some images that do not fulfil statistical assumptions underlying DCP, the results may be faulty.

14.4.2 Comparison to Polarization Based Reconstruction

A second test is conducted to compare our results to a physically accurate, polarization based method [Schechner and Karpel, 2005]. Two images used in that work are processed with our method (images 1 and 2 on Fig. 14.6). Since the results presented in [Schechner and Karpel, 2005] are also further processed by correcting with white balance, the same post-processing is also applied to our method. The effects are depicted in Fig. 14.8 and 14.9.

The results achieved with [Schechner and Karpel, 2005] outperform the dark channel based approach. There are fewer artefacts and the overall contrast is higher. That being said, our results show significant improvement over the raw images and does not need any special procedure when capturing the image. Despite superior results, changing polarization filter during capture is impossible in most real life underwater applications.

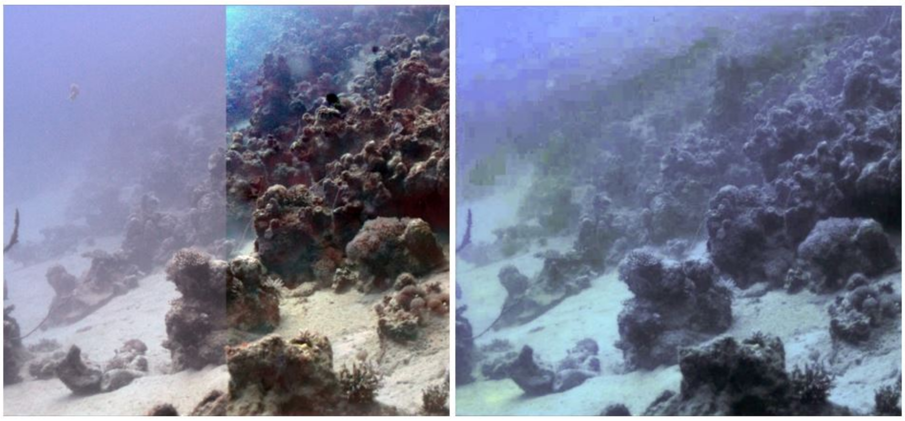


Figure 14.8: Image corrected with [Schechner and Karpel, 2005] (left) and our results (right).

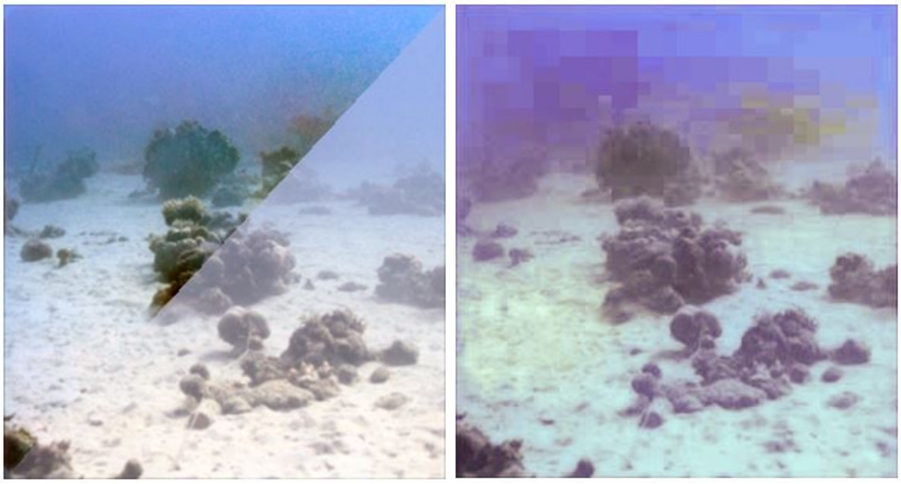


Figure 14.9: Image corrected with [Schechner and Karpel, 2005] (left) and our results (right).

Part V

APPLICATIONS

Chapter 15

DexROV Hardware Selection and System Design

In this chapter, the application of the algorithm described in Part III is presented. The work was motivated by the needs of the DexROV project. Vision system meeting high accuracy requirements had to be designed. Additional constraints were created by the maximum baseline limited by the ROV size and the task given. Object recognition and manipulation requires dense 3D reconstruction.

15.1 Requirements

15.1.1 Hardware Constraints

- Stereo system baseline limited to 0.5m by ROV size. This constraint already takes into account a reserve for the size of the housings and mounting elements.

- Operating at the depth of 1500m. This constraint does not influence the design of the stereo system, only the housings. However, it is a part of the specification and is included here for the sake of completeness.

15.1.2 Hard Performance Requirements

- Accuracy within given range: 1cm standard deviation 2m away from the camera and 15 cm standard deviation at the distance of 5m, as specified in deliverable 2.2. When estimating the accuracy of the system during the design stage a maximum error is usually computed. Therefore $\pm 1.5\sigma$ will be taken as a measure for maximum error allowed. Distances of 2m and 5m away where chosen as expected distances for recognition and navigation respectively. I.e. we want to guarantee that the system will provide reliable navigational data as early as 5m away from the obstacle and will be able to perform at least initial recognition 2m away.

- Minimum observing distance: 30cm. This constraint guarantees that it will be possible to use the stereo system in manipulation tasks.

15.1.3 Soft Performance Requirements

- 3D reconstruction of the observed environment that is as dense as possible.

- Frame rate > 10 fps. Usually, lowering the resolution allows for increasing frame rate (bandwidth is the limiting factor). The 10fps constraint arises from our experience in performing similar tasks. However, the exact value will only be specified following initial tests. In general, the higher the frame rate, the better.

15.2 Implementation and Adaptation to Underwater Conditions

The algorithm was implemented in Matlab. There are two scripts in the package. The first is responsible for the design process as described in Sec. 12.2-12.4. The second script allows for analysing the chosen setup as presented in Sec. 12.5. In the first step, β is fixed and equal to 120°. It can be set as any value, but the actual reconstruction of the object's surface depends on many other factors: light conditions, texture, reconstruction method etc. Therefore, even though $\beta = 120°$ may seem large, the part of the surface that can actually be reconstructed is smaller. As it is only used for providing an initial suggestion for the final selection, it is also not that important. This was also discussed in detail in Sec. 12.6. When designing a stereo system for underwater applications, refraction effects need to be taken into consideration. A detailed description of this phenomenon and methods for handling them were already discussed in Part II. In the DexROV project, Pinax is applied and housings with flat front glass panes are used. Therefore, the perceived image is magnified. This needs to be taken into account at the design step. Fortunately, this magnification may be easily estimated. As discussed in Sec. 6, the change in focal length is equal to the ratio of the water to air refraction index. Therefore, it may be easily estimated by increasing this parameter for each lens by 30%. With this adaptation the remaining part of the algorithm may be applied as usual.

15.3 Application Example

The DexROV project required the design of a stereo system that could be reliably used for both navigation and supervising manipulation tasks performed by the ROV. It was decided that Fire Wire cameras will be used. To limit the choice even further only cameras offered by Point Grey are taken into account. This manufacturer was

Table 15.1: List of sensor models and most important parameters

Model	Resolution	Pixel size [μm]
Sony ICX414 CCD	648 x 488	9.9
Sony ICX274 CCD	1624 x 1224	4.4
Sony ICX818 CCD	1928 x 1448	3.69
Sony ICX267 CCD	1384 x 1032	4.65
Sony ICX687 CCD	1928 x 1448	3.69
Sony ICX674 CCD	1932 x 1452	4.54
Sony ICX694 CCD	2736 x 2192	4.54
Sony ICX808 CCD	2016 x 2016	3.1
Sony ICX285 CCD	1384 x 1036	6.45
Sony ICX625 CCD	2448 x 2048	3.45
Sony ICX825 CCD	1384 x 1032	6.45

chosen as we have good experience in cooperating with them. The list of sensors taken into consideration is presented in the table 15.1.

Since the system is supposed to support navigation of the vehicle, it was decided that a focal length no longer than 5.5mm in air will be used to maintain a wide field of view. For the same reason, a non-verged design was selected. This guarantees the field of view of the system big enough for the task given. To begin with, a general analysis was performed as described in Sec. 12.2.3. A summary of this analysis is presented in Tab. 15.2

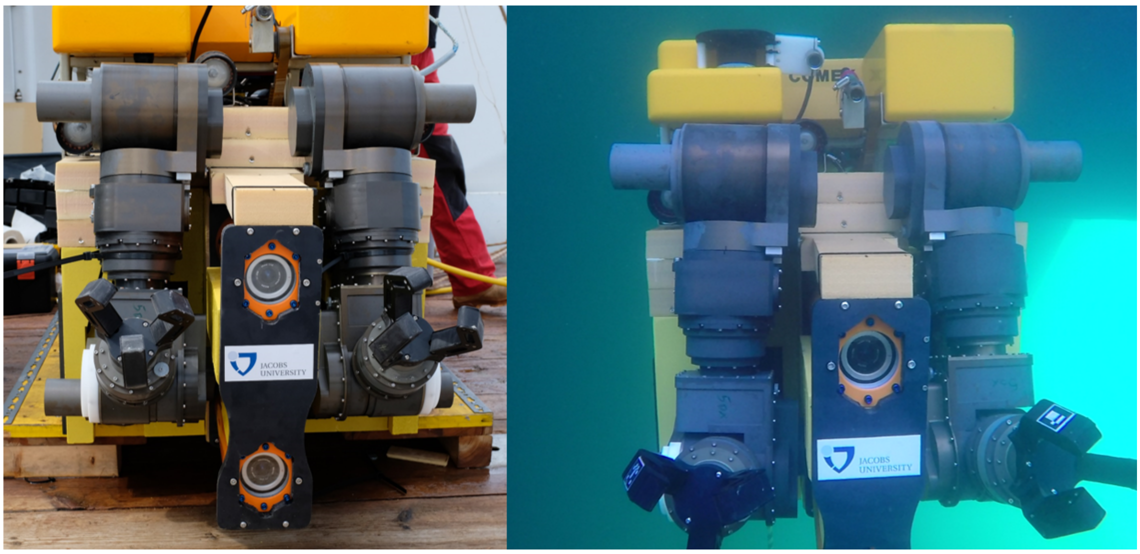


Figure 15.1: DexROV stereo system. Left: integrated and ready to work on the ship's deck. Right: in use, underwater.

This provided very important clues on how to design the system. Finally, Point Grey Grasshopper2 1394b cameras were selected (with Sony ICX285 CCD sensors, and lenses with a focal length of 4.8mm). A baseline of $0.3m$ was chosen. This setup was then analysed as described in Sec. 12.5 (see the results in Fig. 15.2).

The obtained results meet the requirements of the project, so it was decided to use them in the final setup. This concludes the design stage. The system was manufactured and assembled in accordance to the optimized parameters (Fig.15.1). It was also successfully deployed and used during DexROV trials, Marseille 2017.

Table 15.2: Table of results given by the algorithm

Focal lengths / Sensors	2.8 mm	3.8 mm	4 mm	4.8 mm	5.5 mm
Sony ICX414 CCD	More cameras required	More cameras required	0.335 /0.34641 m	0.28 /0.34641 m	0.245 /0.34641 m
Sony ICX274 CCD	0.21 /0.34641 m	0.155 /0.34641 m	0.15 /0.34641 m	0.125 /0.34641 m	0.11 /0.34641 m
Sony ICX818 CCD	0.18 /0.34641 m	0.13 /0.34641 m	0.125 /0.34641 m	0.105 /0.34641 m	0.09 /0.34641 m
Sony ICX267 CCD	0.225 /0.34641 m	0.165 /0.34641 m	0.16 /0.34641 m	0.13 /0.34641 m	0.115 /0.34641 m
Sony ICX687 CCD	0.18 /0.34641 m	0.13 /0.34641 m	0.125 /0.34641 m	0.105 /0.34641 m	0.09 /0.34641 m
Sony ICX674 CCD	0.22 /0.34641 m	0.16 /0.34641 m	0.155 /0.34641 m	0.13 /0.34641 m	0.115 /0.34641 m
Sony ICX694 CCD	0.22 /0.34641 m	0.16 /0.34641 m	0.155 /0.34641 m	0.13 /0.34641 m	0.115 /0.34641 m
Sony ICX808 CCD	0.15 /0.34641 m	0.11 /0.34641 m	0.105 /0.34641 m	0.09 /0.34641 m	0.08 /0.34089 m
Sony ICX285 CCD	0.31 /0.34641 m	0.23 /0.34641 m	0.22 /0.34641 m	0.18 /0.34641 m	0.16 /0.34641 m
Sony ICX625 CCD	0.165 /0.34641 m	0.125 /0.34641 m	0.12 /0.34641 m	0.1 /0.34641 m	0.085 /0.34641 m
Sony ICX825 CCD	0.31 /0.34641 m	0.23 /0.34641 m	0.22 /0.34641 m	0.18 /0.34641 m	0.16 /0.34641 m

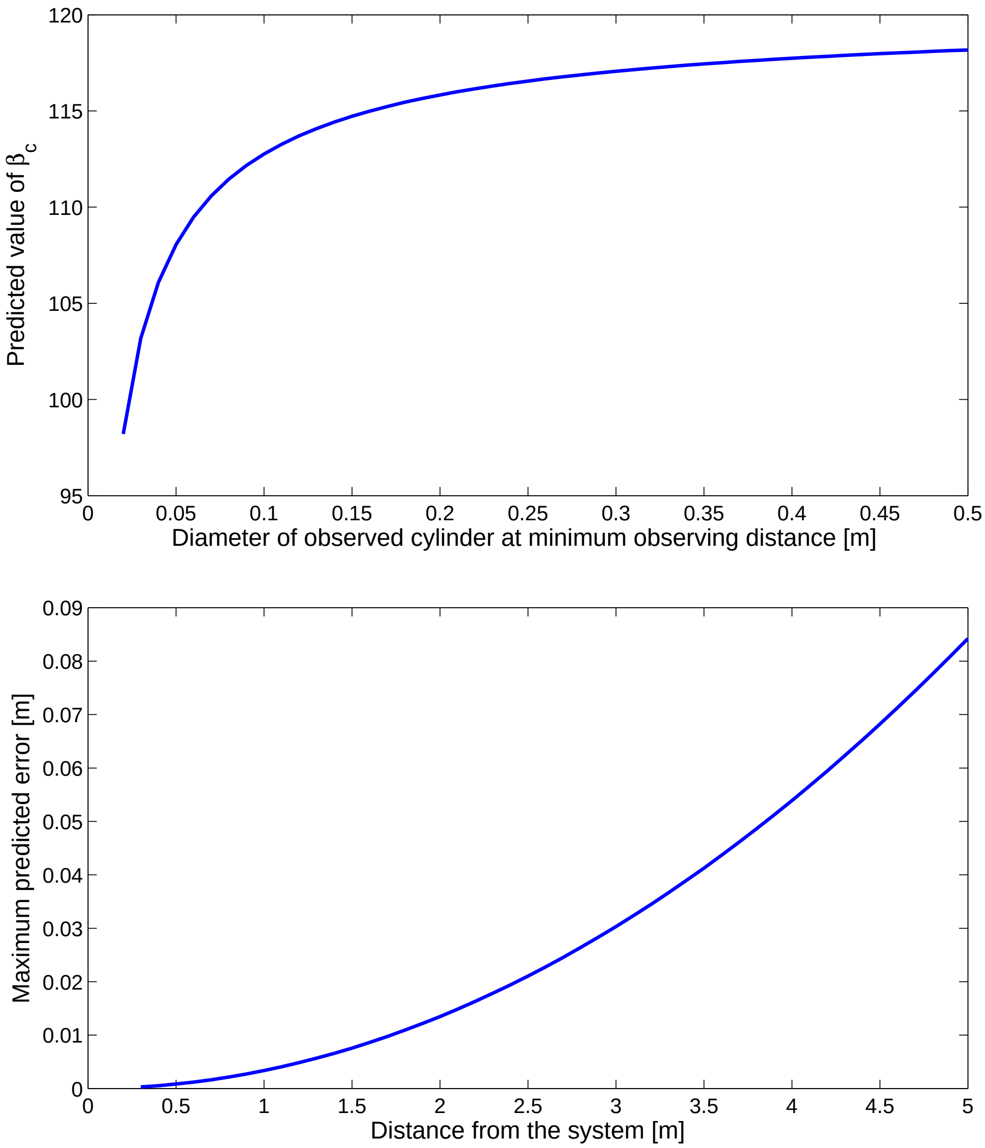


Figure 15.2: The results of system analysis for: Sony ICX285 CCD sensor, 4.8mm lens, $B = 0.3m$, $d' = 0.3m$, no vergence. Top: reconstruction capabilities as a function of the diameter of the cylinder placed at the minimum observing distance. Bottom: predicted accuracy of the reconstruction at given distances.

Chapter 16

Pinax Applications

Pinax model was implemented and the code was released. Two implementations are provided: Matlab and C/C++ ROS packages. Code examples and some test data are available on GitHub: https://github.com/tomluc/Pinax-camera-model .

16.1 Matlab Examples

16.1.1 Script #1: Finding the optimal d_0^* distance

Open *Optimal_d_0/Main.m*. Adjust setup parameters: camera intrinsic matrix K, glass thickness d_1, water and glass refraction indices n_w and n_g. Run the script: ${}^{p}d_0^* =$ *optimal_phisical_d_0* [mm], ${}^{v}d_0^* =$*virtual_d_0* [mm].

16.1.2 Script #2: Calculating the correction maps

To run this example mex opencv is required (https://github.com/kyamagu/mexopencv). Code responsible for analytical forward projection was provided by authors of [Agrawal et al., 2012]. To see the example open and run *Find_correction_map/FindMap.m.* Remember to adjust all the camera information and setup parameters. As an output two files are created: *MapX.txt* and *MapY.txt*. These can be used for image correction, e.g. with opencv *remap(...)* function (compare C/C++ examples). There is also a test image loaded, remapped and saved.

16.2 Robot Operating System C/C++ Examples

Example code consists of the following packages:

- *jir_refractive_image_geometry_msgs* and *jir_refractive_image_geometry*: support packages with the definition of message type used in our processing pipeline and some support functions.

- *jir_rectification_remap_lib*: library providing image correction given correction maps.
- *defraction_map_finder*: package used to find correction maps, given calibration information and setup parameters.
- *jir_image_remapper*: example of node correcting refracted image images.

16.2.1 Example of use

As an example let's see the full process of correcting the images with our method.

1. Calibrate your stereo system. In our work we used camodocal [Heng et al., 2013,Heng et al., 2014,Heng et al., 2015]. This allowed for using more complex camera/lens distortion models. Parts of camodocal code were reused by us when implementing camera models (check license headers in the source code files). Resulting calibration files should be saved (e.g. *camera_left.yaml*).
2. Calculate correction maps for underwater usage. Run *defraction_map_finder* node in *defraction_map_finder* package. This will produce *correctionMap.yaml.* Save it in preferred location.
3. Play included bag file with sample image. Use - *-loop* option.
4. Run *image_remapper.launch* file. Remember to specify the path to the *correctionMap.yaml.* Corrected image is published on the topic specified in the launch file.

16.3 FishGUI

As an answer to the need for collecting data on fish size and population in the Azores region a set of tools based on Pinax model was developed. This task was performed within the MORPH project and was called the FishGUI. Measuring fish is a tedious and difficult task that may be impossible to perform without highly autonomous system. Current methods base on "educated guess" of trained diver estimating the size (and population in the case of observing a school) of fish. Other state of the art method utilizes video assistance, but the fish need to be at the known distance from the camera, orthogonally to the optical axis. This makes it hard to use in most real life conditions.

Proposed solution consists of stereo camera and software developed especially for the purpose of measuring fish. Prototype was built with of the shelf GoPro cameras. Software assisting the scientists allows for loading left and right and processing it to get the disparity image. Then any two valid points on the disparity image may be selected to calculate the distance between them. Measured distance may be labelled,

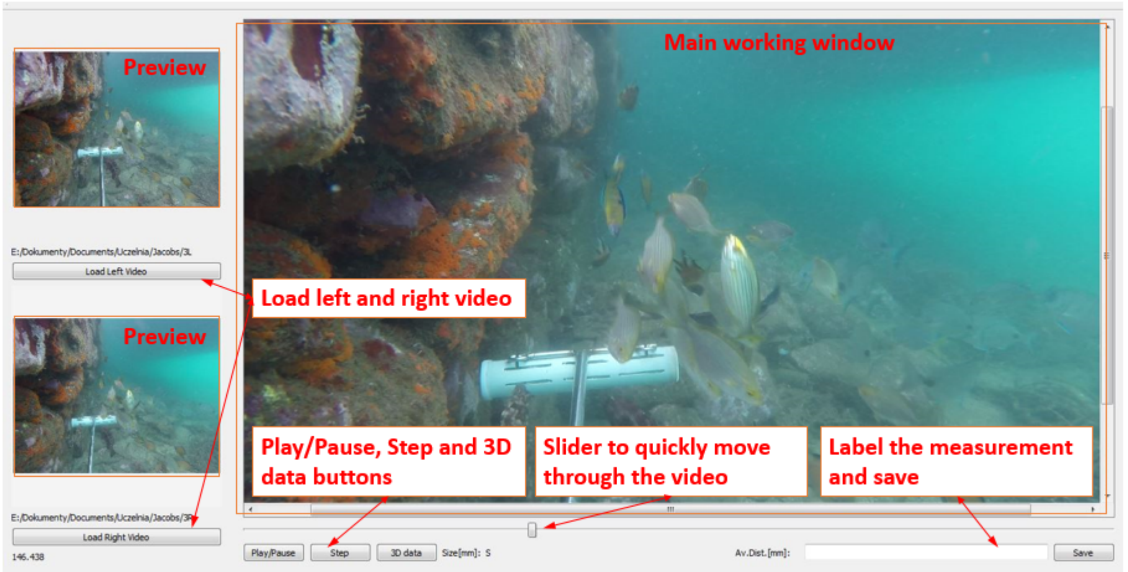


Figure 16.1: FishGUI: graphical user interface utilising Pinax model and stereo reconstruction for measuring fish.

e.g. with the name of the species and saved to a .csv file for further analysis (compare Fig.16.1).

Hardware and software tools discussed here were used in real life conditions. One of the companies responsible for the air transport of live fish needed to estimate the total biomass of a shipment of fish very sensitive to manipulation (pilot fish) in order to optimize transport conditions (volume, O_2, etc.). The goal was to reduce the mortality rate. As reported by our partners from Marine and Environmental Sciences Centre and University of Azores this was a success.

Chapter 17

3D Grid Map Transmission Optimization

The DexROV project, as described in Sec. 1.3, requires a significant amount of data to be transmitted over the satellite link. Among other tasks the surroundings of the ROV must be modelled and the model compressed before the transmission. Part III discussed the design process of the perception system. Thanks to findings described in Part II the environment may be reliably modelled in 3D. This chapter describes usage of the 3D grid map for efficient modelling and transmission of this data. The main goal of this research is to minimize the bandwidth usage, while preserving maximum information about the environment. These results have been published in the proceedings of the IEEE Oceans'17 conference, titled *"3d grid map transmission for underwater mapping and visualization under bandwidth constraints"* ([Łuczyński et al., 2017a]).

17.1 Map Representation

The ROV generates a 3D map based on an underwater stereo system with accurate calibration using the Pinax model (see Part II) with the help of navigation sensors. The mapping task may be performed in multiple ways from odometry using simple sensors to advanced self-localization and mapping techniques, allowing to trade map quality with required computation power and update rates. However this lies beyond the scope of this thesis and will not be discussed here.

Once the map has been built, an efficient representation is required to store the 3D map on the ROV for further navigation and as basis for transmission to the MCC. It was decided that 3D grid map will be used. The octree data-structure [Meagher, 1980] is well known to be a very efficient basis for a 3D grid map representation [Meagher, 1982] and the OctoMap [Hornung et al., 2013] software is a widely used implementation of this data-structure, which is also used here. The octree provides several beneficial properties, which are exploited in this work for the efficient data transmission under bandwidth constraints. A major advantage is its

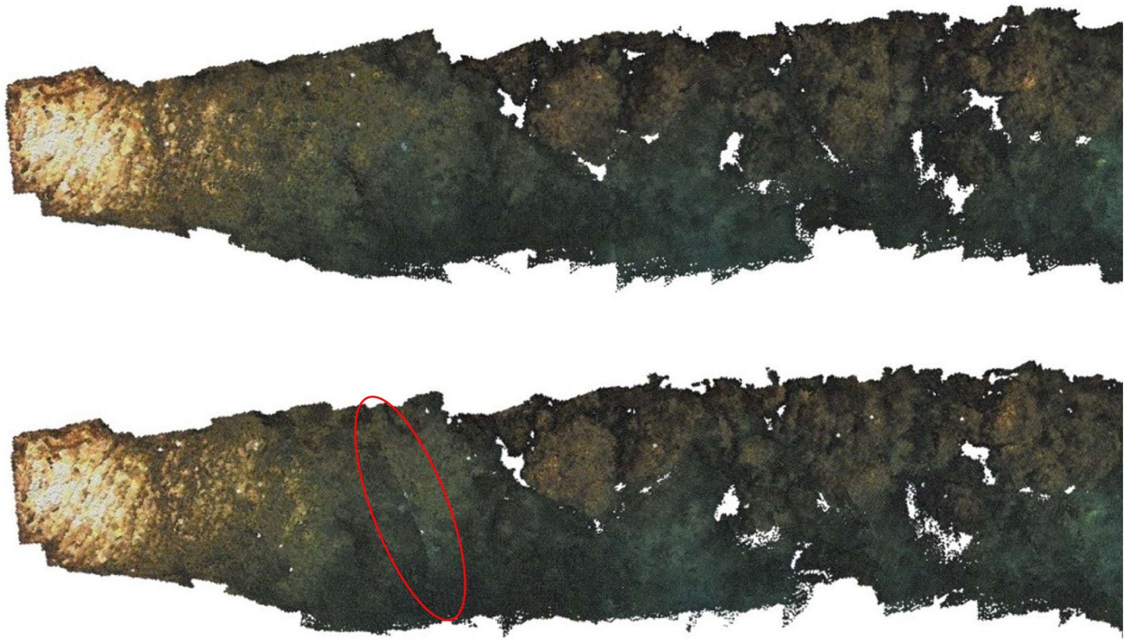


Figure 17.1: Top: OctoMap coloured with our approach. Bottom: naive implementation, last registered point decides about the colour of the given voxel. Please note that our method eliminates the sharp colour changes on the edges of registered scans (e.g. marked with a red loop). It also improves the overall appearance and colour accuracy when recording inconsistently illuminated environment.

storage efficiency, which is a major criterion with respect to the given bandwidth constraints. This representation is further modified to improve the transmission efficiency to the MCC. Furthermore, a colour enhancement of the map is introduced by exploiting the OctoMap representation. The details of this modifications and the whole process will be discussed in the following sections.

17.2 Colour Representation

Underwater images suffer from exceptionally bad light conditions. Not only is there often a low amount of natural light in the scene, but the present light is heavily attenuated and scattered. Description of the underwater image formation model is provided in Chapter 13.

Colour deterioration may be partially reduced when using an OctoMap. Each voxel, which is the smallest data unit used in an OctoMap, accumulates multiple observations of the same region. These measurements may differ in perceived colour. This gives the possibility of selecting the most accurate colour to create a map with overall more accurate colours than those in the registered images.

Theoretically, when a constant ambient light illuminates the whole scene equally, colours from the shortest distance should be taken. However there is a more efficient way of achieving the same result: when multiple points fall into the same voxel the brightest colour is selected to be stored in a voxel. With this simple measure we guarantee that the map colours are taken from points when the camera was close to the surface and/or the additional lights (if available) were in the optimal position to properly illuminate the given patch of the surface.

In order to present the results of the proposed method for assigning colours to the voxels, the following experiment has been conducted, producing the results of Fig. 17.1: The same data was accumulated in the OctoMap twice. The first run was performed using the method proposed here and the second time with a naive implementation – a voxel takes the colour of the last point falling into the given voxel. The data used here was recorded in Biograd na Moru in Croatia in 2015 within the MORPH project. The AUV was moving near the surface recording the seabed. It is easy to observe that there are sharp lines on the edges between the subsequent stereo scans, whereas in the colour-optimized OctoMap these differences are not visible (compare Fig. 17.1).

17.3 OctoMap Transmission

As discussed earlier, representing the environment using an octree has multiple advantages. On the other hand, this also creates some challenges for the DexROV application scenario. After each registration of a new point cloud from stereo, an updated map would have to be transmitted to the Mission Control Center to give the operator the best knowledge about the environment. Of course this is not acceptable from the perspective of bandwidth limitations. Very often, once the map has been built, new point clouds typically only change a small part of the 3D map.

It is important to notice at this point that from the perspective of the MCC the OctoMap structure per se is not important either. ROV operators make the decisions based on the visualization, but the navigation happens internally on board of the vehicle. Therefore sending the full OctoMap after registering each new point cloud is not only impossible from the bandwidth limitation perspective but also unnecessary. Furthermore, for the majority of the map, higher-level voxels (with a low resolution) can just be transmitted as in many regions high resolution is not needed. Only in certain regions specified by the operator the maximum resolution is necessary.

This allows for the following operation scheme:

- The OctoMap is constructed with the maximum possible resolution (with respect to the on-board computer computation capabilities).
- Every time a new point cloud is registered, the OctoMap is locally updated.
- When integrating the point cloud into the map, it is checked if any voxel has changed its state, i.e., if a previously unknown region is now identified as free/occupied, a voxel that was occupied is now free or the free voxel is now occupied.

- The user has an option to specify the bounding box for a region of interest. Voxels in this region will be transmitted with highest available resolution and with colour, voxels outside this bounding box will be transmitted with lower resolution and without colour. Following this rule, voxels that have changed their state are added to the list of voxels that will be published with adequate colour information and resolution.
- When all the points from the given point cloud are integrated into the OctoMap, four lists of voxels are being published: newly-occupied voxels with coarse resolution, newly-occupied voxels within the bounding box using fine resolution and the respective lists of newly-unoccupied voxels.

This way, in the beginning of the mission every new region is transmitted to the operator with low resolution. After this first stage of the mission, when the ROV flies over already known terrain, no data is transmitted unless high resolution for some region is needed or the environment has changed. At the same time the log odds in the OctoMap are constantly updated on board of the vehicle. Of course the MCC may request the transmission of the full OctoMap with highest possible resolution. This operation will take more time but is only triggered when online visualization is not needed, e.g. after completing the mission, for the documentation, or to (re)plan the mission.

17.4 Effectiveness Validation in Simulation

17.4.1 Scene Modeling

Within the DexROV framework, a simulation environment has been developed and utilized in a continuous integration manner [Fromm et al., 2017]. The system includes a simulated ROV equipped with sensors as well as simulated textured environment.

In order to allow for the operator to quickly perceive the current vehicle view, occupied areas, and autonomously recognized objects, a simulated depth camera beneath a simulated RGB camera was used (Fig. 17.2(c) & (e)) to autonomously recognize and localize objects in the scene (Fig. 17.2(f)) using different texture and shape-based machine recognition methods [Bülow and Birk, 2013] [Mueller and Birk, 2016]. The same RGBD input is used to generate the OctoMap, which is used for autonomous collision avoidance and coarse-grained scene representation (Fig. 17.2(d)).

With the help of this testbed, it is possible to evaluate proposed method with ground truth data.

17.4.2 Bandwidth Constraint Modeling

In order to simulate the bandwidth constraints imposed on the scenario software components are deployed in one Docker [Merkel, 2014] container each for the ROV,

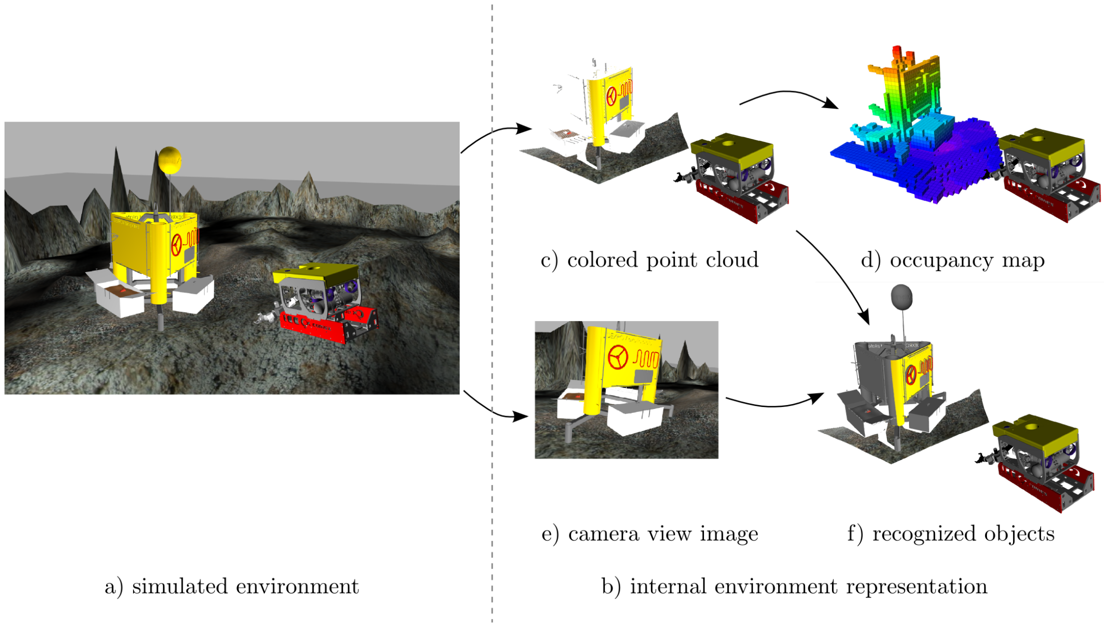


Figure 17.2: Scene Modelling Overview, reprinted from [Fromm et al., 2017]

the operator vessel and the MCC stations. These containers are interfaced with a network simulator [Pfingsthorn, 2016] based on network emulation tool *netem* which allows for dynamic changes of several parameters:

- available bandwidth
- delay (with variance)
- package loss percentage
- package corruptness percentage.

Setting these parameters on realistic limitations reflects the actual system in a realistic way.

17.4.3 Results

Experiments were conducted as follows: stereo point clouds and navigation data are recorded in the simulator. The ROV is teleoperated to first explore a small region of the seabed and then move around this already known environment. This scenario allows to evaluate the bandwidth usage when building a map and during update of an existing one. Furthermore, as mentioned in Sec. 17.3, it is possible to use non-coloured and low-resolution voxels in some regions of the map; during this experiment, however, the whole map was treated as a priority region and transmitted with the highest possible resolution and with colour. The bandwidth used for this transmission is compared to transmitting the same map as a full OctoMap – each time a new scan is integrated the full OctoMap is transmitted.

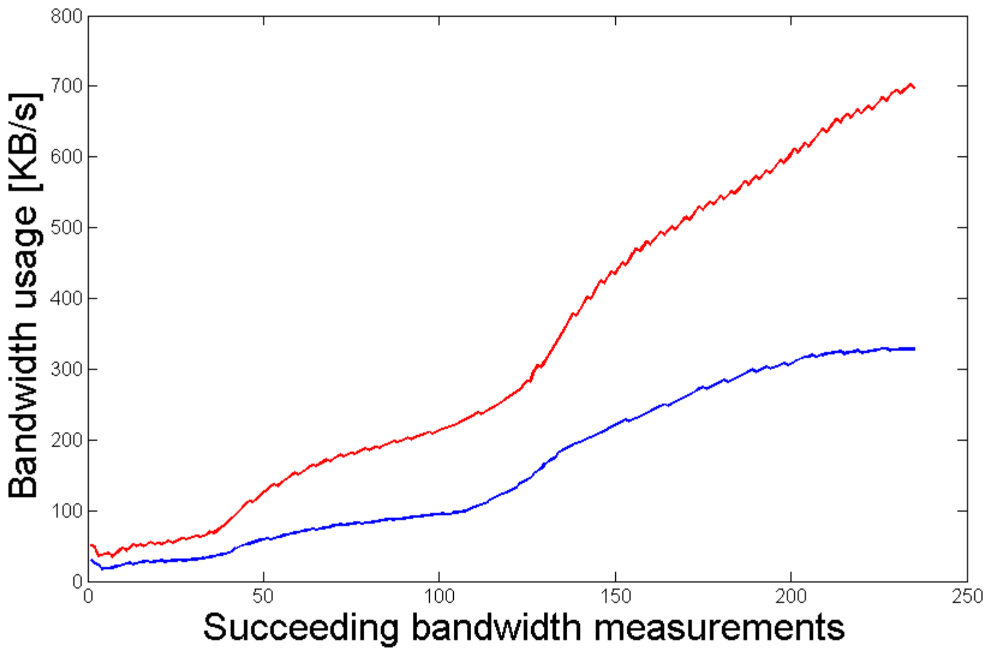


Figure 17.3: Comparison of bandwidth usage for 10 (red) and 15 (blue) cm resolution OctoMap. In both cases the full OctoMap was transmitted after each scan (classic transmission scheme).

This experiment was conducted twice on the same dataset: once with the OctoMap working on 10 cm resolution and the second time on a 15 cm resolution. Fig. 17.3 shows how the bandwidth usage is changing over time, when transmitting the full OctoMap for both resolutions. Fig. 17.4 shows a comparison of the same scenarios, only this time using our proposed method. Finally Fig. 17.5 presents a direct comparison of our method and a transmission of the full OctoMap using 15 cm resolution.

The incremental update obviously clearly outperforms the continuous retransmission of the map. More importantly, the overall implementation is suited to be used in the context of remote ROV control under high bandwidth constraints.

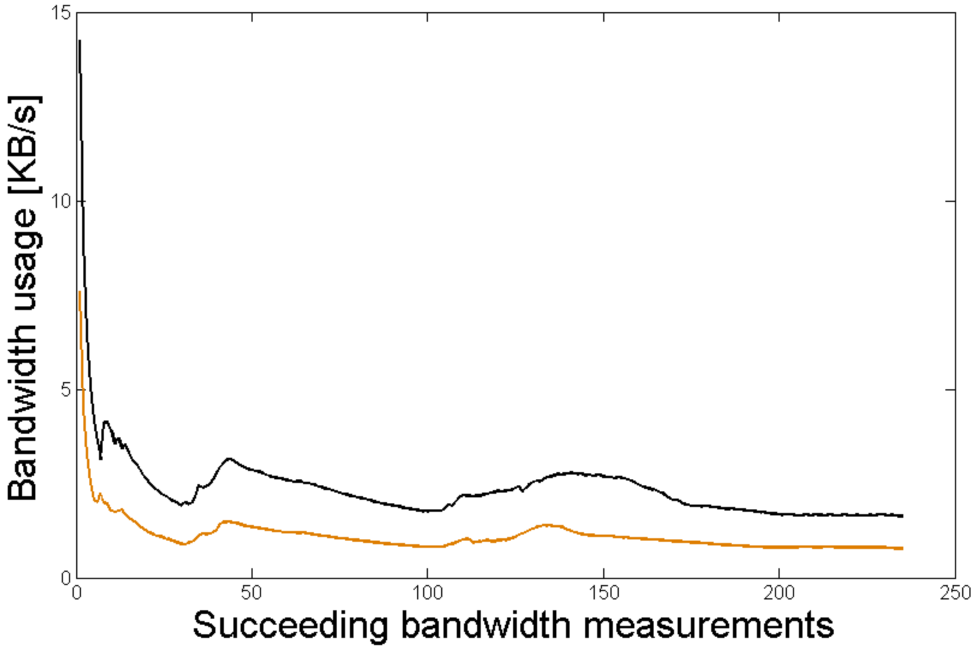


Figure 17.4: Comparison of bandwidth usage for 10 (black) and 15 (orange) cm resolution OctoMap. In both cases the our transmission scheme was used.

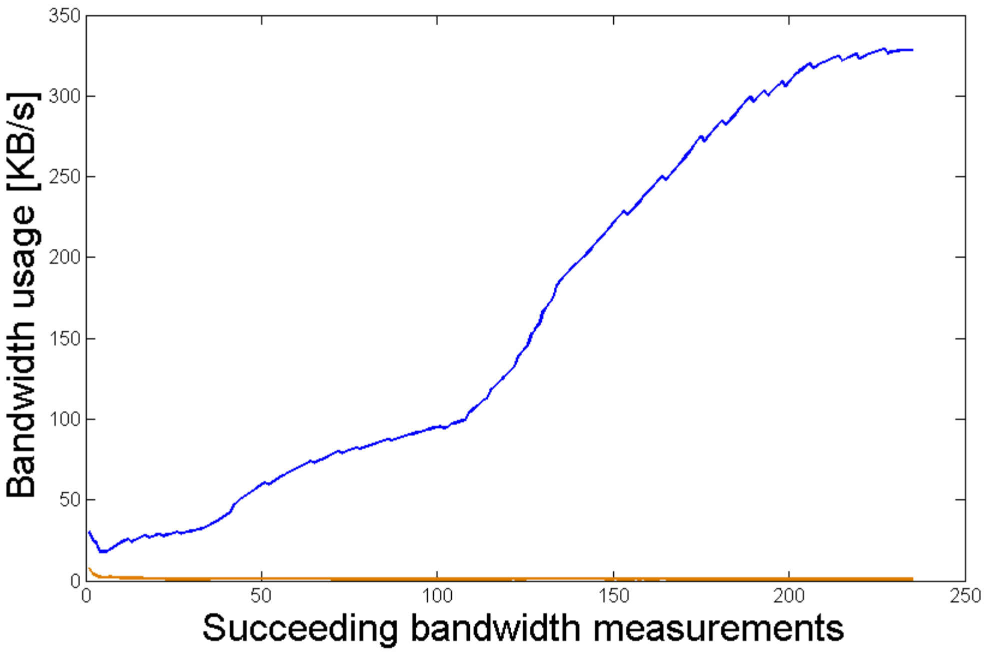


Figure 17.5: Direct comparison of our method (orange) and transmission of the full OctoMap (blue) using 15cm resolution.

Part VI

CONCLUSIONS

Chapter 18

Summary of Contributions

In this thesis an underwater vision system was analysed on every stage of development: from hardware selection through calibration to the processing of the recorded data. Being motivated by real needs of multiple research projects results presented here are robust and were tested in various conditions, serving multiple applications. Contributions to the scientific community presented in this thesis answer the research questions stated in Section 1.4 and are shortly summarized below.

18.1 New Underwater Camera Model

A new camera model, called Pinax, was proposed. This model is based on a virtual pinhole camera model - which was demonstrated to be applicable for real world underwater housings where the camera is relatively close to the flat-pane - while using the projection function of an axial camera. In practice this lead to simplifying the calibration significantly - no underwater procedure is necessary. In-air calibration followed by some calculations is enough to get a valid, underwater pinhole representation. This allows for using a wide range of computer vision algorithms, developed for dry environment, underwater. Especially it is possible to apply closed-form stereo reconstruction algorithms. Removing the need of performing in situ underwater calibration procedure also reduces significantly operational costs for such a mission.

18.2 Stereo System Design Algorithm

An analysis of the 3D reconstruction error was provided: what causes this error, how it may be modelled and some ways of minimizing it. This analysis was then used to develop an algorithm for selecting the cameras, lenses and optimal position for them in the stereo setup. This contribution is important from the practical perspective as it allows for improving the stereo 3D results already at the level of designing the system, even before buying any hardware.

18.3 Haze Removal from Underwater Images

Investigating the underwater image formation process lead to the conclusion that backscatter, responsible for fogginess of the image, is the main factor reducing the 3D reconstruction capabilities. A new method of enhancing the image quality, based on the Dark Channel Prior, was proposed. This contribution showed good performance and promising results, when tested on real underwater data.

18.4 Other

Apart from the main contributions summarized before a range of applications was presented, demonstrating that work discussed in this thesis is not only valid but also has a potential of making a significant impact on the practical underwater computer vision processing. For example a vision system designed in accordance to the method presented in Part III was calibrated with the Pinax camera model and the data generated by this system was used to build a map of the underwater environment. Later on, as this was required by the conditions of the project, this map was successfully compressed to be transmitted over the satellite link with significant bandwidth limitations.

Bibliography

[Agrawal et al., 2012] Agrawal, A., Ramalingam, S., Taguchi, Y., and Chari, V. (2012). A theory of multi-layer flat refractive geometry. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3346–3353.

[Asakawa et al., 2000] Asakawa, K., Kojima, J., Kato, Y., Matsumoto, S., and Kato, N. (2000). Autonomous underwater vehicle aqua explorer 2 for inspection of underwater cables. In *Underwater Technology, 2000. UT 00. Proceedings of the 2000 International Symposium on*, pages 242–247.

[Bahrami and Goncharov, 2010] Bahrami, M. and Goncharov, A. V. (2010). All-spherical catadioptric telescope design for wide-field imaging. *Appl. Opt.*, 49(30):5705–5712.

[Beall et al., 2011] Beall, C., Dellaert, F., Mahon, I., and Williams, S. B. (2011). Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys. In *OCEANS, 2011 IEEE - Spain*.

[Bingham et al., 2010] Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., and Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):702–717. TY - JOUR.

[Bodenmann et al., 2013] Bodenmann, A., Thornton, B., Nakajima, R., Yamamoto, H., and Ura, T. (2013). Wide area 3d seafloor reconstruction and its application to sea fauna density mapping. In *Oceans - San Diego, 2013*.

[Brandou et al., 2007] Brandou, V., Allais, A., Perrier, M., Malis, E., Rives, P., Sarrazin, J., and Sarradin, P. (2007). 3d reconstruction of natural underwater scenes using the stereovision system iris. In *OCEANS 2007 - Europe*.

[Brignone et al., 2011] Brignone, L., Munaro, M., Allais, A., and Opderbecke, J. (2011). First sea trials of a laser aided three dimensional underwater image mosaicing technique. In *OCEANS*. IEEE.

[Bülow and Birk, 2013] Bülow, H. and Birk, A. (2013). Spectral 6-DOF Registration of Noisy 3D Range Data with Partial Overlap. *Transactions on Pattern Analysis and Machine Intelligence*, 35(4):954–969.

[Chapman et al., 2010] Chapman, P., Bale, K., and Drap, P. (2010). We all live in a virtual submarine. *Computer Graphics and Applications, IEEE*, 30(1):85–89.

[Chari and Sturm, 2009] Chari, V. and Sturm, P. F. (2009). Multi-view geometry of the refractive plane. In *BMVC*. British Machine Vision Association.

[Chen et al., 2007] Chen, J., Khatibi, S., and Kulesza, W. (2007). Planning of a multi stereo visual sensor system - depth accuracy and variable baseline approach. In *2007 3DTV Conference*.

[Chen and Yang, 2014] Chen, X. and Yang, Y.-H. (2014). Two-view camera housing parameters calibration for multi-layer flat refractive interface. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 524–531.

[Cheng et al., 2015] Cheng, C. Y., Sung, C. C., and Chang, H. H. (2015). Underwater image restoration by red-dark channel prior and point spread function deconvolution. In *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 110–115.

[Chiang and Chen, 2012] Chiang, J. Y. and Chen, Y. C. (2012). Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing*, 21(4):1756–1769.

[Davie et al., 2008] Davie, A., Hartmann, K., Timms, G., de Groot, M., and McCulloch, J. (2008). Benthic habitat mapping with autonomous underwater vehicles. In *OCEANS 2008*.

[Desouza and Kak, 2002] Desouza, G. N. and Kak, A. C. (2002). Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267.

[Dubbelman and Groen, 2009] Dubbelman, G. and Groen, F. C. A. (2009). Bias reduction for stereo based motion estimation with applications to large scale visual odometry. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2222–2229.

[Enchev et al., 2015] Enchev, I., Pfingsthorn, M., Łuczyński, T., Sokolovski, I., Birk, A., and Tietjen, D. (2015). Underwater place recognition in noisy stereo data using fab-map with a multimodal vocabulary from 2d texture and 3d surface descriptors. In *OCEANS 2015 - Genoa*.

[Foresti, 2001] Foresti, G. (2001). Visual inspection of sea bottom structures by an autonomous underwater vehicle. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 31(5):691–705.

[Fromm et al., 2017] Fromm, T., Mueller, C. A., Pfingsthorn, M., Birk, A., and di Lillo, P. (2017). Efficient Continuous System Integration and Validation for Deep-Sea Robotics Applications. In *OCEANS 2017 - Aberdeen*.

[Galceran et al., 2014] Galceran, E., Campos, R., Palomeras, N., Carreras, M., and Ridao, P. (2014). Coverage path planning with realtime replanning for inspection of 3d underwater structures. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6586–6591.

[Gancet et al., 2015] Gancet, J., Urbina, D., Letier, P., Ilzokvitz, M., Weiss, P., Gauch, F., Antonelli, G., Indiveri, G., Casalino, G., Birk, A., Pfingsthorn, M. F., Calinon, S., Tanwani, A., Turetta, A., Walen, C., and Guilpain, L. (2015). Dexrov: Dexterous undersea inspection and maintenance in presence of communication latencies. *IFAC-PapersOnLine*, 48(2):218 – 223. 4th IFAC Workshop onNavigation, Guidance and Controlof Underwater VehiclesNGCUV 2015.

[Gedge et al., 2011] Gedge, J., Gong, M., and Yang, Y.-H. (2011). Refractive epipolar geometry for underwater stereo matching. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 146–152.

[Gracias and Santos-Victor, 2000] Gracias, N. and Santos-Victor, J. (2000). Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding*, 79(1):66–91.

[Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision.* Cambridge University Press, Cambridge.

[He et al., 2011] He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353.

[Heng et al., 2014] Heng, L., Burki, M., Lee, G. H., Furgale, P., Siegwart, R., and Pollefeys, M. (2014). Infrastructure-based calibration of a multi-camera rig. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4912–4919.

[Heng et al., 2015] Heng, L., Furgale, P. T., and Pollefeys, M. (2015). Leveraging image-based localization for infrastructure-based calibration of a multi-camera rig. *J. Field Robotics*, 32(5):775–802.

[Heng et al., 2013] Heng, L., Li, B., and Pollefeys, M. (2013). Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1793–1800.

[Herath et al., 2006] Herath, D. C., Kodagoda, K. R. S., and Dissanayake, G. (2006). Modeling errors in small baseline stereo for slam. In *2006 9th International Conference on Control, Automation, Robotics and Vision.*

[Hollinger et al., 2012] Hollinger, G., Englot, B., Hover, F., Mitra, U., and Sukhatme, G. (2012). Uncertainty-driven view planning for underwater inspection. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4884–4891.

[Horgan and Toal, 2006] Horgan, J. and Toal, D. (2006). Review of machine vision applications in unmanned underwater vehicles. In *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*.

[Hornung et al., 2013] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*. Software available at http://octomap.github.com.

[Hue et al., 2011] Hue, J., Serayet, M., Drap, P., Papini, O., and Wuerbel, E. (2011). *Underwater Archaeological 3D Surveys Validation within the Removed Sets Framework Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 6717 of *Lecture Notes in Computer Science*, pages 663–674. Springer Berlin / Heidelberg.

[Jaffe, 1990] Jaffe, J. S. (1990). Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering*, 15(2):101–111.

[Johnson-Roberson et al., 2010] Johnson-Roberson, M., Pizarro, O., Williams, S. B., and Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51.

[Jordt et al., 2016] Jordt, A., Köser, K., and Koch, R. (2016). Refractive 3d reconstruction on underwater images. *Methods in Oceanography*, 15:90 – 113. Computer Vision in Oceanography.

[Jordt-Sedlazeck et al., 2013] Jordt-Sedlazeck, A., Jung, D., and Koch, R. (2013). *Refractive Plane Sweep for Underwater Images*, pages 333–342. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Jordt-Sedlazeck and Koch, 2012] Jordt-Sedlazeck, A. and Koch, R. (2012). Refractive calibration of underwater cameras. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 846–859.

[Jordt-Sedlazeck and Koch, 2013] Jordt-Sedlazeck, A. and Koch, R. (2013). Refractive structure-from-motion on underwater images. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 57–64.

[Jr et al., 2013] Jr, P. D., do Nascimento, E., Moraes, F., Botelho, S., and Campos, M. (2013). Transmission estimation in underwater single images. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 825–830.

[Kalwa et al., 2012] Kalwa, J., Pascoal, A., Ridao, P., Birk, A., Eichhorn, M., and Brignone, L. (2012). The european rnd-project morph: Marine robotic systems of self-organizing, logically linked physical nodes. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV)*.

[Kang et al., 2012] Kang, L., Wu, L., and Yang, Y.-H. (2012). Experimental study of the influence of refraction on underwater three-dimensional reconstruction using the svp camera model. *Appl. Opt.*, 51(31):7591–7603.

[Kawahara et al., 2013] Kawahara, R., Nobuhara, S., and Matsuyama, T. (2013). A pixel-wise varifocal camera model for efficient forward projection and linear extrinsic calibration of underwater cameras with flat housings. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 819–824.

[Kim and Eustice, 2013] Kim, A. and Eustice, R. (2013). Real-time visual slam for autonomous underwater hull inspection using visual saliency. *Robotics, IEEE Transactions on*, 29(3):719–733.

[Kunz and Singh, 2008] Kunz, C. and Singh, H. (2008). Hemispherical refraction and camera calibration in underwater vision. In *OCEANS 2008*.

[Kunz and Singh, 2010] Kunz, C. and Singh, H. (2010). Stereo self-calibration for seafloor mapping using auvs. In *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*.

[Lavest et al., 2003] Lavest, J.-M., Rives, G., and LaprestÃ©, J.-T. (2003). Dry camera calibration for underwater applications. *Mach. Vis. Appl.*, 13(5):245–253.

[Levin et al., 2008] Levin, A., Lischinski, D., and Weiss, Y. (2008). A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242.

[Li et al., 1997] Li, R., Li, H., Zou, W., Smith, R., and Curran, T. (1997). Quantitative photogrammetric analysis of digital underwater video imagery. *Oceanic Engineering, IEEE Journal of*, 22(2):364–375.

[Lots et al., 2000] Lots, J. F., Lane, D. M., and Trucco, E. (2000). Application of 2 1/2 d visual servoing to underwater vehicle station-keeping. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 2, pages 1257–1264 vol.2.

[Łuczyński and Birk, 2017] Łuczyński, T. and Birk, A. (2017). Underwater image haze removal with an underwater-ready dark channel prior. In *OCEANS 2017 - Anchorage*.

[Łuczyński et al., 2017a] Łuczyński, T., Fromm, T., Govindaraj, S., Mueller, C., and Birk, A. (2017a). 3d grid map transmission for underwater mapping and visualization under bandwidth constraints. In *OCEANS 2017 - Anchorage*.

[Łuczyński et al., 2017b] Łuczyński, T., Pfingsthorn, M., and Birk, A. (2017b). Image rectification with the pinax camera model in underwater stereo systems with verged cameras. In *OCEANS 2017 - Anchorage*.

[Łuczyński et al., 2017c] Łuczyński, T., Pfingsthorn, M., and Birk, A. (2017c). The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings. *Ocean Engineering*, 133:9 – 22.

[Mallios et al., 2011] Mallios, A., Ridao, P., Carreras, M., and Hernandez, E. (2011). Navigating and mapping with the sparus auv in a natural and unstructured underwater environment. In *OCEANS 2011*.

[Marks et al., 1994] Marks, R. L., Wang, H. H., Lee, M. J., and Rock, S. M. (1994). Automatic visual station keeping of an underwater robot. In *OCEANS*, volume 2, pages II/137–II/142 vol.2. IEEE.

[Matthies and Shafer, 1987] Matthies, L. and Shafer, S. (1987). Error modeling in stereo navigation. *IEEE Journal on Robotics and Automation*, 3(3):239–248.

[McLeod et al., 2013] McLeod, D., Jacobson, J., Hardy, M., and Embry, C. (2013). Autonomous inspection using an underwater 3d lidar. In *Oceans - San Diego, 2013*.

[Meagher, 1980] Meagher, D. (1980). Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. Technical report, Rensselaer Polytechnic Institute.

[Meagher, 1982] Meagher, D. (1982). Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147.

[Merkel, 2014] Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal*, 2014(239).

[Mišković et al., 2016] Mišković, N., Bibuli, M., Birk, A., Caccia, M., Egi, M., Grammer, K., Marroni, A., Neasham, J., Pascoal, A., Vasilijevi, A., and Vuki, Z. (2016). Caddycognitive autonomous diving buddy: Two years of underwater human-robot interaction. *Marine Technology Society Journal*, 50(4):54–66.

[Millard and Seaver, 1990] Millard, R. C. and Seaver, G. (1990). An index of refraction algorithm for seawater over temperature, pressure, salinity, density, and wavelength. *Deep Sea Research Part A. Oceanographic Research Papers*, 37(12):1909–1926.

[Mueller and Birk, 2016] Mueller, C. A. and Birk, A. (2016). Hierarchical Graph-Based Discovery of Non-Primitive-Shaped Objects in Unstructured Environments. In *International Conference on Robotics and Automation*.

[Nayar, 1997] Nayar, S. K. (1997). Catadioptric omnidirectional camera. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 482–488.

[Negahdaripour et al., 2006] Negahdaripour, S., Barufaldi, C., and Khamene, A. (2006). Integrated system for robust 6-dof positioning utilizing new closed-form visual motion estimation methods in planar terrains. *Oceanic Engineering, IEEE Journal of*, 31(3):533–550.

[Negahdaripour and Firoozfam, 2006] Negahdaripour, S. and Firoozfam, P. (2006). An rov stereovision system for ship-hull inspection. *Oceanic Engineering, IEEE Journal of*, 31(3):551 –564.

[Negahdaripour and Fox, 1991] Negahdaripour, S. and Fox, J. (1991). Undersea optical stationkeeping: Improved methods. *Journal of Robotic Systems*, 8(3):319–338.

[Negahdaripour et al., 2007] Negahdaripour, S., Sekkati, H., and Pirsiavash, H. (2007). Opti-acoustic stereo imaging, system calibration and 3-d reconstruction. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*.

[Pessel et al., 2003] Pessel, N., Opderbecke, J., and Aldon, M.-J. (2003). Camera self-calibration in underwater environment. In *11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*.

[Pfingsthorn, 2016] Pfingsthorn, M. (2016). *Docker Networking Simulation*. Jacobs University. https://github.com/maxpfingsthorn/mini-network-simulator.

[Pizarro et al., 2003] Pizarro, O., Eustice, R., and Singh, H. (2003). Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing Techniques and Applications*, pages 601–612.

[Pojar et al., 2012] Pojar, D., Jeong, P., and Nedevschi, S. (2012). Robust visual odometry using stereo reconstruction error model. In *Intelligent Computer Communication and Processing (ICCP), 2012 IEEE International Conference on*, pages 149–154.

[Purcell et al., 2011] Purcell, M., Gallo, D., Packard, G., Dennett, M., Rothenbeck, M., Sherrell, A., and Pascaud, S. (2011). Use of remus 6000 auvs in the search for the air france flight 447. In *OCEANS 2011*.

[Quan and Fry, 1995] Quan, X. and Fry, E. S. (1995). Empirical equation for the index of refraction of seawater. *Applied Optics*, 34(18):3477–3480.

[Ramalingam et al., 2006] Ramalingam, S., Sturm, P., and Lodha, S. (2006). Theory and calibration algorithms for axial cameras. In Narayanan, P., Nayar, S. K., and Shum, H.-Y., editors, *Asian Conference on Computer Vision, ACCV 2006, January, 2006*, volume 3851 of *Lecture Notes in Computer Science*, pages 704–713, Hyderabad, Inde. Springer.

[Rodriguez and Aggarwal, 1990] Rodriguez, J. J. and Aggarwal, J. K. (1990). Stochastic analysis of stereo quantization error. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):467–470.

[Roswell et al., 1976] Roswell, W., Halikas, A., and Halikas, G. (1976). The index of refraction of seawater - technical report. *University of California, San Diego Visibility Laboratory*.

[Sandwell et al., 2014] Sandwell, D. T., Müller, R. D., Smith, W. H. F., Garcia, E., and Francis, R. (2014). New global marine gravity model from cryosat-2 and jason-1 reveals buried tectonic structure. *Science*, 346(6205):65–67.

[Schechner and Averbuch, 2007] Schechner, Y. Y. and Averbuch, Y. (2007). Regularized image recovery in scattering media. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1655–1660.

[Schechner and Karpel, 2005] Schechner, Y. Y. and Karpel, N. (2005). Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of Oceanic Engineering*, 30(3):570–587.

[Schettini and Corchs, 2010] Schettini, R. and Corchs, S. (2010). Underwater Image Processing: State of the Art of Restoration and Image Enhancement Methods. *EURASIP Journal on Advances in Signal Processing*, 2010(1):746052.

[Sedlazeck and Koch, 2011] Sedlazeck, A. and Koch, R. (2011). Calibration of Housing Parameters for Underwater Stereo-Camera Rigs. In Hoey, J., editor, *Proceedings of the British Machine Vision Conference*. BMVA Press. http://dx.doi.org/10.5244/C.25.118.

[Sedlazeck et al., 2009] Sedlazeck, A., Koser, K., and Koch, R. (2009). 3d reconstruction based on underwater video from rov kiel 6000 considering underwater imaging conditions. In *OCEANS 2009 - EUROPE*.

[Servos et al., 2013] Servos, J., Smart, M., and Waslander, S. (2013). Underwater stereo slam with refraction correction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3350–3355.

[Shortis and Harvey, 1998] Shortis, M. R. and Harvey, E. S. (1998). Design and calibration of an underwater stereo-video system for the monitoring of marine fauna populations. *International Archives of Photogrammetry and Remote Sensing*, 32(1).

[Sturm et al., 2006] Sturm, P., Ramalingam, S., and Lodha, S. (2006). *Imaging Beyond the Pinhole Camera*, chapter On Calibration, Structure from Motion and Multi-View Geometry for Generic Camera Models, pages 87–105. Springer.

[Treibitz et al., 2012] Treibitz, T., Schechner, Y., Kunz, C., and Singh, H. (2012). Flat refractive geometry. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):51–65.

[Treibitz and Schechner, 2009] Treibitz, T. and Schechner, Y. Y. (2009). Active polarization descattering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):385–399.

[Treibitz et al., 2008] Treibitz, T., Schechner, Y. Y., and Singh, H. (2008). Flat refractive geometry. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.*

[Uhlmann, 1995] Uhlmann, J. K. (1995). *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford Oxford.

[Voss, 1991] Voss, K. J. (1991). Simple empirical model of the oceanic point spread function. *Appl. Opt.*, 30(18):2647–2651.

[Yamashita et al., 2007] Yamashita, A., Fujii, M., and Kaneko, T. (2007). Color registration of underwater images for underwater sensing with consideration of light attenuation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 4570–4575.

[Yamashita et al., 2011] Yamashita, A., Kawanishi, R., Koketsu, T., Kaneko, T., and Asama, H. (2011). Underwater sensing with omni-directional stereo camera. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 304–311.

[Yau et al., 2013] Yau, T., Gong, M., and Yang, Y.-H. (2013). Underwater camera calibration using wavelength triangulation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2499–2506.

[Zhang and Boult, 2011] Zhang, T. and Boult, T. (2011). Realistic stereo error models and finite optimal stereo baselines. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 426–433.

[Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334.

[Zhang et al., 2016] Zhang, Z., Rebecq, H., Forster, C., and Scaramuzza, D. (2016). Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 801–808.