Retina tracking for robot-assisted vitreoretinal surgery

Y.G.M. Douven
CST 2015.008
Abstract

In vitreoretinal surgery – surgery on the back of the eye – surgeons benefit from robotic assistance. An important step during robotic assistance is tracking the back of the eye. However, during surgery, an eye can move in its socket and current visualization techniques cannot view the entire retina in one shot. Hence, tracking the position and the shape of the retina with high accuracy is difficult. This has two main reasons. First, only part of the retina may be visible, e.g., when the retina moves out of the field of view, or when the field of view is temporarily obstructed. Second, visually distinguishable features on the retina will move relative to each other when the eye deforms. There is currently no method available that is accurate enough for mapping and tracking the retina in three dimensions and in real-time for application in robotic assistance. The contributions of this thesis cover five aspects of retina tracking for robot-assisted vitreoretinal surgery. First, a retina visualization modality is selected that can measure the retina in real-time and in three dimensions and can serve as a good input for algorithms that map and track the retina. Second, a notion is presented on how to convert the measurements from the visualization output into useful data for a mapping and tracking algorithm. Third, a mapping and tracking algorithm is derived that allows for tracking of temporarily invisible parts of the retina. Fourth, a mapping and tracking algorithm is proposed that integrates a deformation-dependent model of the retina, allowing to improve the accuracy of the map. Fifth, an eye-pressure dependent model of the eye is derived that is implemented in the mapping and tracking algorithm. The proposed algorithm is implemented in C++ and validation experiments are executed using a balloon as a model of the retina. Both simulations and experiments validate and demonstrate the ability of the developed mapping and tracking algorithm to accurately track parts of the retina that are temporarily invisible and estimate the expansion of the eye. They show a significant improvement in accuracy compared to basic filtering.
# Contents

1 Introduction 4  
1.1 Medical robotics and eye surgery ................................................. 4  
1.2 Previous research in the field of retina tracking for information augmentation .......................... 4  
1.3 Goal .................................................................................... 5  
1.4 Approach and contributions .......................................................... 5  
1.5 Report outline ...................................................................... 6  

2 Visualization modality selection for retina tracking 7  
2.1 Evaluation of retinal visualization modalities in general eye examination ............................ 8  
  2.1.1 Visualization modality I: fundus photography .................................................... 8  
    2.1.1.1 Fundus photography method I: infrared imaging ........................................... 8  
    2.1.1.2 Fundus photography method II: fluorescent angiography .......................... 8  
    2.1.1.3 Fundus photography method III: autofluorescence .................................... 9  
  2.1.2 Visualization modality II: confocal scanning laser ophthalmoscopy (cSLO) ............... 9  
  2.1.3 Visualization modality III: ultrasound biomicroscopy (UBM) ............................... 9  
  2.1.4 Visualization modality IV: optical coherence tomography (OCT) .......................... 9  
  2.1.5 Visualization modality V: indirect ophthalmoscopy ........................................... 10  
  2.2 Visualization modality selection for retina tracking .................................................. 10  
    2.2.1 Visualization modalities used in a surgical setting ........................................... 10  
    2.2.2 Optimal visualization modality: confocal scanning laser ophthalmoscopy .......... 10  
    2.2.3 Selected visualization modality: stereo-vision ................................................. 11  
    2.2.4 Extension to selected modality: OCT integration and robotic encoder data ........ 11  

3 Observer-based SLAM and a model with interconnected landmarks for the retina 13  
3.1 Introduction ........................................................................ 14  
3.2 Observer-based SLAM .......................................................... 15  
  3.2.1 Definitions and standard SLAM ........................................................................ 15  
  3.2.2 Observer-based SLAM ............................................................................. 15  
3.3 Modeled landmark interconnections .................................................. 16  
  3.3.1 Extending the state transition of the EKF ....................................................... 16  
  3.3.2 Influence on state estimation ......................................................................... 17  
3.4 Model of the eye .................................................................... 17  
  3.4.1 Assumptions and model properties ............................................................... 17  
  3.4.2 Implementation .............................................................................. 17  
  3.4.3 Polynomial approximation ........................................................................... 18  
  3.4.4 Solving the quadratic problem ....................................................................... 18  
  3.4.5 Augmented EKF ..................................................................................... 19  
3.5 Simulation results ................................................................... 19  
  3.5.1 Observer-based SLAM ............................................................................... 19  
  3.5.2 Modeled landmark interconnections ............................................................. 20  
3.6 Conclusions and recommendations .................................................. 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Implementation and experimental results</td>
<td>22</td>
</tr>
<tr>
<td>4.1 Experimental setup</td>
<td>22</td>
</tr>
<tr>
<td>4.2 Ground truth</td>
<td>23</td>
</tr>
<tr>
<td>4.3 Results</td>
<td>23</td>
</tr>
<tr>
<td>4.3.1 Experiment 1: tracking of temporarily invisible points</td>
<td>23</td>
</tr>
<tr>
<td>4.3.2 Experiment 2: estimating change in expansion</td>
<td>24</td>
</tr>
<tr>
<td>4.4 Conclusions</td>
<td>26</td>
</tr>
<tr>
<td>5 Conclusions and recommendations</td>
<td>27</td>
</tr>
<tr>
<td>Appendixes</td>
<td>30</td>
</tr>
<tr>
<td>A Implementation of observer-based SLAM with modeled landmark interconnections</td>
<td>30</td>
</tr>
<tr>
<td>A.1 Initial map and feature list</td>
<td>30</td>
</tr>
<tr>
<td>A.2 Update map with new features</td>
<td>30</td>
</tr>
<tr>
<td>A.3 Update existing features with new measurements</td>
<td>31</td>
</tr>
<tr>
<td>A.4 Possible observer update: Iterative Closest Point</td>
<td>32</td>
</tr>
<tr>
<td>B Real-time three-dimensional object detection in vitreoretinal surgery</td>
<td>33</td>
</tr>
<tr>
<td>B.1 Undistorting and rectifying camera images</td>
<td>33</td>
</tr>
<tr>
<td>B.2 Stereo-vision</td>
<td>33</td>
</tr>
<tr>
<td>B.3 Tool recognition</td>
<td>33</td>
</tr>
<tr>
<td>B.4 Retina tracking</td>
<td>34</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Medical robotics and eye surgery

Medical robotics is an interdisciplinary field that focuses on developing mechatronic devices for diagnosis and therapy. The long-term goal of this area is to enable new medical techniques and to improve the outcome of existing procedures by enhancing the capabilities to the physician. From a technical viewpoint, most of the commercially available surgical robot systems used in clinical practice are not much more than very sophisticated tools; all the decision-making is carried out by surgeons. Sometimes the surgeons are aided by diagnostic and imaging tools during the pre-operative planning phase, or by displaying relevant information during the intervention. Robotic systems can enhance the surgeon's ability. Realisation of this requires embedding feedback inside the robot controller. This feedback can be regarded as cognitive assistance, which facilitates both substantial safety improvements for the patient and medical staff, including surgeon, and a significant improvement of performance including usability of the system. Such cognitive assistance can help and assist the surgeon in the decision making process and in the execution of the targeted procedures.

PRECEYES is a spin-off company of the Eindhoven University of Technology, developing the PRECEYES Surgical System [1], see Figure 1.1. The system provides robotic assistance for vitreoretinal eye surgery, i.e., surgery at the retina and the fluids inside the eye. Vitreoretinal surgery is regarded as one of the most delicate types of surgery and can only be performed by the very best of the eye surgeons. Nevertheless, the precision of these surgeons is limited, thus limiting the development of new procedures. The PRECEYES Surgical System improves the precision of a surgeon 10 to 20 times. This facilitates improving the procedure time, which is beneficial for the patients, and for the surgeon's comfort, which enables the surgeon to do a better job. Moreover, the improved precision enables the development of new procedures treating patients that cannot be treated manually at this moment [2][3].

The current version of the PRECEYES Surgical System consists of a motion controller and an instrument manipulator, using various filters and controlling techniques for reducing hand tremor and improving precision. It does not yet provide cognitive assistance, as it has no feedback from the environment. A future version of the system will provide safety and performance improvements by including vision-based feedback that supports the surgeon in performing vitreoretinal surgical procedures. To be able to provide vision-based feedback, the surgical system needs to be able to receive visual feedback from the retina. It can then map and track the retina and provide useful feedback.

1.2 Previous research in the field of retina tracking for information augmentation

Consider an eye during surgery. Over time it can move in its socket. Tracking the position and the shape of the retina with high accuracy is difficult, as I) only part of the retina may be visible, e.g., when the retina moves out of the field of view, or when the field of view is temporarily obstructed, and II) visually distinguishable features on the retina will move relative to each other when the eye deforms. There is currently no method available for accurately mapping and tracking the retina in three dimensions and in real-time for application in robotic assistance. There are research programs
focusing on retina mapping and tracking [4][5][6]. They are used to provide information overlay to surgeons, improving navigation during surgery, and focus on two-dimensional mapping and tracking. The researches can basically be divided into two groups. The first group focuses on hybrid tracking and mosaicking of the retina [4][5]. In real-time, images of small part of the retina are combined and over time form an image of a larger part of the retina, called mosaicking. The second group is developing methodology that is based on template matching between preoperative and intra-operative images. It is already being applied in eye examination for aligning multiple examination visualization modalities [6]. Intra-operative retina tracking is still in experimental phase and only used for information augmentation for the surgeon.

1.3 Goal

The goal of this master thesis is twofold: I) compile a retinal visualization modality selection for three-dimensional retina measurements, and II) develop a three-dimensional retina mapping and tracking algorithm. The visualization modality selection should be integrated in the current surgical setting and should hence be minimally invasive to the proximity of the surgeon. Last, a large viewing angle on the retina is required. Since the algorithm is first introduced in a test phase, cost should also be low. The mapping and tracking algorithm should be able to perform in real-time, be stable and robust against measurement noise and visual obstructions.

Accuracy is the most important part of retina tracking, since it facilitates safety improvements and an improvement in performance. The accuracy mostly depends on the resolution of the measurement devices in the visualization modality, and should be as high as possible within reasonable price.

Selecting a visualization modality leads to a number of problems. Does the current surgical visualization setup, which consists of a stereo microscope, meet the requirements? Is this setup optimal, or is there a better solution? Finally a suitable way to validate the mapping and tracking algorithm needs to be found.

1.4 Approach and contributions

For the development of a retinal mapping and tracking method, vitreoretinal surgery is considered. During vitreoretinal surgery, a clear view on the retina is desirable. A visualization modality selection is made that can measure the retina in three dimensions. To meet the space and cost requirements, the current surgical setting has been investigated for applicability and optimality. It is compared with other retina visualization modalities used in general eye examination. A visualization modality that is
optimal for real-time three-dimensional measuring of the retina is chosen, but will not be used, since it will have to drastically change the current surgical setting. A final modality is selected that functions as input data for retinal vein junction recognition software [7]. The junctions that are found serve as input for the mapping and tracking algorithm. This software will not be discussed in this report, and the output will hence otherwise be created. Some ideas on the matter are presented in Appendix B, a result from a Bachelor Thesis.

Various mapping algorithms are compared and a suitable method is selected and adapted for applicability in the eye. The final mapping method must be able to track parts of the retina that are temporarily invisible and must be able to cope with the deformations of the retina. Finally a simplified model of the tissue of the eye is developed to improve the accuracy of the map. The resulting algorithm is demonstrated in simulations and experiments, and compared to the existing mapping technique it is based on.

The contributions of this master thesis are fivefold. First, a visualization modality selection has been made that can measure the retina in real-time and in three dimensions. Second, an idea has been given on how to convert the measurements from the visualization output into useful data for a mapping and tracking algorithm. Third, a mapping and tracking algorithm is derived that allows for tracking of temporarily invisible parts of the retina. Fourth, a mapping and tracking algorithm is proposed that integrates a deformation-dependent model of the object, allowing to improve the accuracy of mapping of deformable objects like the eye. Fifth, an eye-pressure dependent model of the eye is derived that is implemented in the mapping and tracking algorithm. Simulation and experimental results are presented, showing the application possibilities of the algorithm and the improved accuracy compared to standard algorithms.

1.5 Report outline

The outline of this report is as follows. In Chapter 2, retina visualization modalities are analyzed and discussed, and a selection is made. In Chapter 3, an adapted mapping algorithm is presented, for mapping and tracking the retina, which is verified with simulations. In Chapter 4, the proposed tracking method is verified with experiments. In Chapter 5, conclusions and recommendations are presented.
Chapter 2

Visualization modality selection for retina tracking

Literature and background

In this chapter, a vision setup is selected that measures the inner eye in three dimensions. To select a feasible visualization setup, a variety of medical imaging methods are compared and space and implementation limitations are discussed and compared. An optimal setup is presented. The selected setup differs from this, as the optimal setup is too expensive and replaces the complete current setup.

Figure 2.1 shows an overview of the most widely used visualization modalities in general eye examination and during surgery. All modalities discussed in this chapter can be found in this overview.

* Diagnosis, also pre- and postoperative
** Photo through microscope
*** As a basis for the B-scan, the A-scan on itself is not used in eye examination

Figure 2.1: Retina visualization modality overview. The dark gray colored block is an optimal visualization modality for three-dimensional retina imaging. The light gray colored blocks denote the selected modalities.
2.1 Evaluation of retinal visualization modalities in general eye examination

The input for any kind of retinal representation relies on imaging techniques measuring the eye. Today, multiple medical imaging techniques are readily available and are more and more being integrated in the operating room. This section presents the 5 most widely used techniques in retinal imaging: different types of cameras used for fundus (or retina) photography, confocal scanning laser ophthalmoscopy, ultrasound biomicroscopy, optical coherence tomography and indirect ophthalmoscopy. Processing the output images provides useful data for creating maps and tracking the retina.

2.1.1 Visualization modality I: fundus photography

The most basic imaging technique is a standard camera, whether or not sensitive to different types of light. Almost all commercially available microscopes have one or more slots to attach a camera. This has two advantages for use in real-time mapping application: no external devices are needed that may be in the way of the surgeon during surgery, and the images recorded by the camera are the same images that are seen be by the surgeon.

The microscopes used during examination and during vitreoretinal surgery are stereo microscopes. These optical microscopes use two separate optical paths with two objectives and eyepieces to provide slightly different viewing angles to the left and to the right eyes. This arrangement produces a three-dimensional visualization of the sample being examined. By attaching a camera to a tapping on each eyepiece, the two resulting camera images can be combined to extract a three-dimensional representation. This is called stereo-vision. Since fundus photography can be performed through the microscope, imaging can be performed non-invasively.

For photographing the fundus, different types of light are used, each for different applications. The three most widely used methods are mentioned below. The resolution and rate of each method depends on the sensor and are proportional to cost. If a reasonable sensor is used, photos can be taken in real-time.

2.1.1.1 Fundus photography method I: infrared imaging

Infrared (IR) imaging uses infrared light rather than white light for illumination [8]. It has several advantages over with light fundus photography:

- Since infrared light is invisible, patients are spared the white light flash that accompanies traditional fundus photography. This is particularly valuable when imaging children and light sensitive patients.

- Infrared light penetrates unclear media better than white light, which is especially helpful when imaging patients with cataracts. This advantage greatly improves the robust for use as input in mapping applications, since correct en full vision on the retina can be guaranteed in more cases.

- IR imaging may offer better visualization of epiretinal membranes (disease of the eye in response to changes in the vitreous humor or more rarely, diabetes) and cystoid macular edema compared to fundus photography and red-free imaging.

2.1.1.2 Fundus photography method II: fluorescent angiography

Intravenous Fluorescein angiography (IVFA) or Fluorescent Angiography (FAG) is a technique for examining the circulation of the retina and choroid using a fluorescent dye and a specialized camera [9][10]. It involves injection of sodium fluorescein into the systemic circulation, and then an angiogram is obtained by photographing the fluorescence emitted after illumination of the retina with blue light at a wavelength of 490 nm. The camera used for photographing can be a standard color CCD, but more often is a monochromatic sensor sensitive to blue light.
2.1.1.3 Fundus photography method III: autofluorescence

Even without the use of dye, parts of the fundus show areas of fluorescence in certain conditions. This so-called autofluorescence is used as a diagnostic indicator and a tool for monitoring disease progression [11]. Fundus autofluorescence (FAF) imaging can provide information about the health and function not just of the central retina but in the periphery as well.

Most systems use blue or green light for excitation. An advantage of longer-wavelength (green) light is that there is less absorption by the crystalline lens of the eye, which is quite autofluorescent with blue light, especially in patients with cataracts.

2.1.2 Visualization modality II: confocal scanning laser ophthalmoscopy (cSLO)

Confocal scanning laser ophthalmoscopy (cSLO) is an ophthalmic imaging technology that uses laser light instead of a bright flash of white light to illuminate the retina [12]. Confocal imaging is the process of scanning an object point by point by a focused laser beam and then capturing the reflected light through a small aperture (a confocal pinhole). The confocal pinhole suppresses light reflected or scattered from outside of the focal plane, which otherwise would blur the image. The result is a sharp, high contrast image of the object layer located at the focal plane.

The advantages of using cSLO over traditional fundus photography include improved image quality, small depth of focus, suppression of scattered light, patient comfort through less bright light, 3D imaging capability, video capability, and effective imaging of patients who do not dilate well. Confocal scanning laser ophthalmoscopy can create up to 20.6 three dimensional images per second. The finest definition is $512 \times 512$ pixels with a maximum angle of view of 30°.

2.1.3 Visualization modality III: ultrasound biomicroscopy (UBM)

Ultrasound biomicroscopy (UBM) is a type of ultrasound eye exam that makes a more detailed image than regular ultrasound [13]. Ultrasound refers to sound waves with a frequency too high for humans to hear. Ultrasound images (sonograms) are made by sending a pulse of ultrasound into tissue using an ultrasound transducer. The sound reflects from parts of the tissue; these reflections are recorded and displayed as an image.

Ultrasound provides real-time cross-sectional images in a very cost-effective manner, even in the presence of optically opaque intervening structures. General medical ultrasonography uses sound waves with a frequency of around 10 Hz. Ultrasound biomicroscopy involves the use of much higher frequencies (35-50 MHz) than those used in conventional ophthalmic B-scanners. The resulting resolution is $30 \ \mu m$ is axial direction, $60 \ \mu m$ in lateral direction.

Currently, a scan rate of 8 Hz can be achieved, giving real-time imaging, with scans consisting of 256 lines-of-sight (vectors) over a 5 mm × 5 mm field.

2.1.4 Visualization modality IV: optical coherence tomography (OCT)

Optical coherence tomography (OCT) is a technology for performing high-resolution cross-sectional imaging [14][15]. OCT is attractive for ophthalmic imaging because image resolutions are 1 to 2 orders of magnitude higher than conventional ultrasound, imaging can be performed non-invasively and in real time, and quantitative morphometric information can be obtained. OCT is somewhat analogous to ultrasound imaging except that it uses light instead of sound. Optical coherence tomography is based on low-coherence interferometry, typically employing near-infrared light. The use of relatively long wavelength light allows it to penetrate into the scattering medium. Confocal microscopy typically penetrates less deeply into the sample but with higher resolution.

A light beam is focused to a point on the surface of the sample under test. Areas of the sample that reflect back a lot of light will create greater interference than areas that don’t. This reflectivity profile, called an A-scan, contains information about the spatial dimensions and location of structures within the sample. A cross-sectional tomograph (B-scan) is achieved by laterally combining a series of these axial depth scans (A-scan). By performing multiple B-scans a three dimensional representation of the sample is created.
Currently, about 29,000 A-scans per second can be performed, resulting in a 170 × 170 image of A-scans with spacial depth information of the sample [16].

2.1.5 Visualization modality V: indirect ophthalmoscopy

With indirect ophthalmoscopy, the fundus is illuminated by light passing through a high-powered positive lens [17][18]. Returning light passes back through the lens to form a real, inverted and laterally reversed image of the fundus between the lens and the practitioner. The difference with direct ophthalmoscopy is the field of view on the retina. With direct ophthalmoscopy, the field of view is smaller, but the magnification is higher.

This method is only for viewing the retina by the physician, not recording it, and it will hence not be further discussed here.

2.2 Visualization modality selection for retina tracking

Changes to the current operation room setup are preferably kept small. This way new techniques like retina mapping and tracking can be implemented and tested without impeding the surgeon during surgery. This implies that the best implementation at this moment will not be the optimal setup for viewing the retina.

2.2.1 Visualization modalities used in a surgical setting

During vitreoretinal surgery, the operating room has typically a setup containing the following elements:

- a stereo microscope
- an operating table with a headrest for the patient attached to it
- a infusion cannula attached to a pressure controller
- a screen with a live feed from one or more cameras that are attached to the microscope

The surgeon is located behind the patient and views the eye through the microscope. To look inside the eye, a BIOM is attached to the microscope. The BIOM (Binocular Indirect Ophthalmo Microscope) incorporates the principle of indirect ophthalmoscopy in the operating microscope and enables up to 120 degree of non-contact observation of the fundus as well as high magnification of the macula area. Under the BIOM, the eye can be rotated freely so that the far periphery of the fundus is also easily viewed.

Due the close proximity of the BIOM to the eye, no space is left for external imaging techniques other than the ones integrated in the microscope. The advantage of viewing through the microscope is that the system will see exactly the same as the surgeon. The disadvantage is the dependency on the surgeon’s ability to focus the microscope to obtain sharp images. If the system would take over this focusing task, any mistake could result in deterioration of the surgeon’s view on the retina, which in turn can lead to damage to the eye.

In laser surgery, a high speed camera is used at close range. The retina is illuminated using infrared light, causing the pupil to light up. The camera then tracks the luminous pupil and uses it to position the laser. This modality can easily be extended with a BIOM so that vision onto the retina becomes possible. It would however have to replace the complete microscope which is not desired.

2.2.2 Optimal visualization modality: confocal scanning laser ophthalmoscopy

Ideally, high resolution three dimensional images of the complete interior of the eye are captured at a high frequency. With the current state of the art, this is not possible. The closest option is a three dimensional video of the retina, viewed through the pupil.
Three dimensional OCT achieves high resolution but image capture times are high when scanning large areas of the fundus. Moreover, extra processing is needed to extract the retinal surface from the depth data. The depth information does however offer extended possibilities when operating on retinal veins, since puncturing depth can be measured.

Three dimensional cSLO only captures the surface of the fundus and does hence not require any extra processing, compared to the depth information from OCT. The image capture time is also shorter and hence it is currently the best solution available for the purpose of tracking the three dimensional retinal surface.

Both modalities require the microscope to be replaced by a scanning device containing dedicated hardware for three dimensional OCT or cSLO. Therefore they are not suited to be applied during intra-operative retina tracking.

2.2.3 Selected visualization modality: stereo-vision

For application in the current surgical setting, a minimally invasive modality is chosen: stereo-vision. The stereo-vision cameras are already present in the operation room, and if not, easy to integrate, since most microscopes have slots for extension parts. It can be used intra operative with minimal changes to the environment, i.e. no parts are in the way. It is therefore also not expensive. Stereo-vision provides a three dimensional image of the environment, but the depth (axial) resolution is much smaller than the lateral (2D image) resolution. Both are dependent on the resolution of the camera sensor (CCD), and hence a good depth resolution requires a very high CCD resolution. This will result in high processing times per image and is therefore not desirable.

A clear view on the retina can be obstructed by numerous causes, like floating tissue or blood in the vitreous. High wavelength light like infrared light penetrates these unclear media better than visual light. Illuminating the retina with infrared light and using infrared CCDs in the cameras will therefore be a possibility that is worth further investigation. This is however beyond the scope of this thesis.

2.2.4 Extension to selected modality: OCT integration and robotic encoder data

For robotic feedback possibilities, a high resolution in all directions is desired. Since the depth resolution of stereo-vision using normal resolution CCDs is not very high, an extension to the modality is required.

The first goal of robotic feedback is providing extra safety to the patient. In vitreoretinal surgery, this mainly means not puncturing the retina. The distance between the surgical tool and the retina is here of the utmost importance. The safest method would be to include a proximity sensor in all surgical tools. This is currently however too expensive to become a standard in hospitals. A less expensive, but almost as safe solution is the implementation of a proximity sensor in the trocar, placed in line with the tool. Combining both the proximity sensor’s information and the stereo images from the microscope would provide an accurate estimate of the retina’s position and a very accurate measurement of the distance between the retina and the surgical instrument. Problems may arise when employing instruments of a shape aberrant from straight. The distance from trocar to retina can still be measured very accurately, but the accuracy drops at the tip of the instrument.

Since OCT penetrates into the tissue, an OCT probe not only provides the system with a proximity measurement, but also extra information is obtained of the underlying tissue. Floating tissue and other small obstructions can then be easily distinguished from the retina. An OCT probe would therefore be an ideal proximity sensor. The micrometer-resolution scans can also be used to find blood vessels. Employing the more recent spectral-domain OCT [19] technique allows for faster scans and hence a higher longitudinal resolution. The use of the a robotic surgical system improves the position estimate of the tool even further. Using the very accurate encoder measurements and combining these with the three-dimensional geometry of the stereo vision microscope cameras and the (SD-)OCT measurements, the system can provide an accurate measurement of the position of the surgical tool compared to the retina.

An overview of the proposed setup is shown in Figure 2.2.
Figure 2.2: Selected retinal visualization modality with OCT extension: a stereo microscope is used to create a three dimensional image of the retina. The OCT probe is integrated in the trocar and is in line with the surgical tool, providing an accurate proximity measurement from the tool to the retinal surface.
Chapter 3

Observer-based SLAM and a model with interconnected landmarks for the retina

In this chapter, a mapping and tracking algorithm for the retina is developed. The output of the selected visualization modality from Chapter 2 can be processed to extract vein junctions [7]. An example of these junctions is shown in Figure 3.1. They serve as input to the mapping and tracking algorithm.

![Vein junctions on a photograph of the retina](image)

Figure 3.1: Vein junctions found on a photograph of the retina. Image from [7]

A model of the eye is derived and integrated in the mapping and tracking algorithm. The result is validated and demonstrated using simulations. Experimental validation is done in Chapter 4.

This chapter is presented in paper form, and is hence incomplete on implementation specifics. Complete explanation on implementation of the algorithm is in Appendix A.
Observer-based SLAM and a model with interconnected landmarks for the retina


Abstract—In this work, a mapping method has been developed for a stationary sensor in a dynamical environment. Simultaneous Localization And Mapping (SLAM) is a method used in mobile robotic applications. It is used to navigate through an unknown environment, while at the same time building a map of the environment. This method has been adapted to be used on a stationary sensor in a dynamical environment, called observer-based SLAM. Objects are identified based on multiple measurement points, which are added to the map. Observer-based SLAM then decouples relative movement of the map features from group movement. In this way, higher accuracy can be achieved, and an estimate can be provided for map features that have not been updated recently.

The extended Kalman filter in the implementation of observer-based SLAM has been extended with a model that exploits a-priori knowledge about the physical interconnection of the map features. This increases the accuracy of the position estimates of the features even further.

The benefits of both contributions are shown in an implementation for eye surgery. For this purpose, a model of the eye – that takes deformations due to changes in eye pressure into account – has been derived and validated in simulations.

Index Terms—Simultaneous Localization And Mapping (SLAM), Extended Kalman Filter (EKF), modeled landmark interconnections, eye pressure estimation, medical robotics, vitreoretinal surgery, retina mapping and tracking

I. INTRODUCTION

MEDICAL robotics is a field developing devices for diagnosis and therapy. The focus is on enhancing capabilities of the physician. From a technical viewpoint, most of the commercially available surgical robot systems are passive instruments. They consist of a master and a slave device that are connected with a bilateral controller, tracking movements of the master device whilst filtering out hand tremor. Most of the decision-making is carried out by surgeons, sometimes aided by diagnostic and imaging tools during the pre-operative planning phase, or by displaying relevant information during the intervention.

Eye surgery is one of the fields that can benefit from robotic assistance, in particular vitreoretinal surgery on the back of the eye. There are no commercially available solutions yet, but some startups are doing relevant research in the field [1][2][3]. In vitreoretinal surgery, imaging is important, which is available through cameras attached to the microscope that is used by the physician to inspect the back of eye. The eye can move and deform, and μm precision is demanded. However, direct sight can be obstructed, i.e. by the surgeon’s hands, operation tools or floating tissue in the eye. As a result, imaging and tracking of the eye for feedback in real-time robotic assistance is difficult. No suitable solution are available yet.

In the field of robotics, simultaneous localization and mapping (SLAM) is used as a standard tool for a mobile sensor moving through a stationary environment [4][5]. SLAM is an excellent starting point for mapping environments whilst estimating the position from which the environment is observed. A map is created by measuring the environment and by detecting objects or “landmarks”. These landmarks are a steady beacon in a further unknown world, and can be used to determine the robot’s own position and the position of newly detected landmarks.

During SLAM, the robot is incrementally conducting distance measurements. Due to sensor noise, at every time increment measurements will have a certain inaccuracy. Any new features being added to the map will contain corresponding errors. This poses a problem, since the errors in the position estimates of the features in the map are mutually dependent [6]. Today’s most widely used algorithms are based on the extended Kalman filter (EKF) [7][8], which solves this problem by estimating a posterior probability distribution.

Consider an eye during surgery. Over time it can move in its socket. Tracking the position and the shape of the retina with high accuracy is difficult, as I) only part of the retina may be visible, e.g., when the retina moves out of the field of view or when the field of view is temporarily obstructed, and II) visually distinguishable features on the retina will move relative to each other when the eye deforms.

In this paper, a SLAM algorithm is derived to create an accurate map of a moving environment, measured from a stationary sensor, which matches the case of an eye during surgery. We call this observer-based SLAM. The EKF in the implementation of observer-based SLAM will be extended with a model that exploits a-priori knowledge about the physical interconnection of the map features. Including this model of the eye increases accuracy decreases sensitivity to measurement noise. Visually distinguishable features on the retina will serve as input to the resulting algorithm.

The contribution of this paper is threefold. First, a SLAM algorithm is derived that enables application of SLAM in the case of a stationary observer and moving objects, as opposed to a moving observer and stationary objects. This allows for tracking of unmeasured landmarks seen at least once. Second, a SLAM algorithm is proposed that integrates a deformation-dependent model of the object, allowing to improve the accuracy of mapping of features on deformable objects. Third, an eye-pressure dependent model of the eye is derived that is implemented in the observer-based SLAM.
algorithm, further increasing the accuracy. Simulation results are presented, showing the application possibilities of this implementation of SLAM and the improved accuracy compared to standard algorithms.

In Section II, observer-based SLAM is introduced. In Section III, the notion of modeled landmark interconnections is presented. In Section IV, the model of the eye is presented and simulation results are presented in Section V.

II. OBSERVER-BASED SLAM

In this section, SLAM is transformed into observer-based SLAM. Observer-based SLAM involves creating a map of a moving environment, measured from a stationary observer, as opposed to standard SLAM, where a stationary environment is measured from a moving observer. An EKF implementation of SLAM is used. The standard odometry update is removed, because there is no incremental measurement of the moving environment for a stationary observer. Then a transformation from local map features to global landmarks is introduced to retrieve the global positions of the measured landmarks.

A. Definitions and standard SLAM

The observer in observer-based SLAM is a stationary sensor. Objects recognized in the sensor data are landmarks and are added to the map as map features. Define the state vector:

\[ \mathbf{x} = [\mathbf{z}; \theta; \mathbf{f}] \]  
(1)

with \( \mathbf{z} \) the observer’s position, \( \theta \) the observer’s orientation and \( \mathbf{f} \) a column vector containing the positions of all map features. \( \mathbf{f} \) has length \( n \times d \), with \( n \) the number of map features and \( d \) the dimension of the environment. \( \mathbf{f} \) is a map of the environment, viewed from the observer’s position.

\( f_i \) is the \( i \)th element in \( \mathbf{f} \), and hence \( \mathbf{f} \) consists of \( f_i \) \( \forall i \in 1, 2, \ldots, n \).

Estimates are denoted with a hat, so \( \hat{\mathbf{x}} \) denotes the estimate of the state \( \mathbf{x} \).

(\( \cdot \)) denotes a variable at iteration \( k \), (\( \cdot \))\(_{k-1} \) denotes the prediction of a variable at iteration \( k \), transitioned from (\( \cdot \))\(_{k-1} \). (\( \cdot \))\(_{k\mid k} \) denotes the estimate of a variable at iteration \( k \).

The EKF is the nonlinear version of the Kalman filter [9] which linearizes about an estimate of the current mean and covariance. The EKF model assumes that the true state at iteration \( k \) is evolved from the state at \( k - 1 \) according to

\[ \mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \]  
(2)

where \( f \) is the state transition model which is applied to the previous state \( \mathbf{x}_{k-1} \) and the input vector \( \mathbf{u}_k \), \( \mathbf{w}_k \) is the process noise, assumed to be a zero mean multivariate Gaussian noise with covariance \( Q_k \). A measurement \( z_k \) of the true state \( \mathbf{x}_k \) is assumed to be made according to

\[ z_k = h(\mathbf{x}_k) + \mathbf{v}_k \]  
(3)

where \( h \) is the observation model which maps the true state space into the observed space and \( \mathbf{v}_k \) is the observation noise which is assumed to be zero mean multivariate Gaussian noise with covariance \( W_k \).

In SLAM, \( f(\mathbf{x}_{k-1}, \mathbf{u}_k) \) is generally used as an odometry update. In case of a full odometry measurement, using (1) this gives:

\[ f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) = f \left( \begin{bmatrix} \hat{\mathbf{z}}_{k-1}^	op; \hat{\theta}_{k-1}^	op; \hat{\mathbf{f}}_{k-1}^	op \end{bmatrix} \right) ; \mathbf{u}_k \]  
(4)

where \( \mathbf{w}_{k,z} \) and \( \mathbf{w}_{k,\theta} \) are the translational and rotational parts of the odometry update at iteration \( k \) respectively.

When the true state satisfies the process described above, at every iteration, the state estimation is updated according to [8]:

\[ \hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k) \]  
(5)

with \( \hat{\mathbf{y}}_k \) the measurement residual (or innovation) defined as:

\[ \hat{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \]  
(6)

and \( \mathbf{K}_k \) the (near-optimal) Kalman gain, defined as:

\[ \mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k \mathbf{S}_k^{-1} \]  
(7)

The predicted covariance estimate \( \mathbf{P}_{k|k-1} \) and the covariance of the innovation \( \mathbf{S}_k \) are defined as:

\[ \mathbf{P}_{k|k-1} = \mathbf{F}_{k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k-1}^\top + \mathbf{Q}_{k-1} \]  
(8)

\[ \mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{W}_k \]  
(9)

where \( \mathbf{F}_{k-1} \) and \( \mathbf{H}_k \) are the state transition and observation matrices defined by:

\[ \mathbf{F}_{k-1} = \frac{\partial f}{\partial \mathbf{x}} \bigg|_{\mathbf{x}_{k-1}; \mathbf{u}_{k-1}} \mathbf{H}_k = \frac{\partial h}{\partial \mathbf{x}} \bigg|_{\hat{\mathbf{x}}_{k|k-1}} \]  
(10)

The predicted state estimate is defined as:

\[ \hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}; \mathbf{u}_{k-1}) \]  
(11)

B. Observer-based SLAM

The basis of observer-based SLAM is the same as normal SLAM. There are two differences between the EKF in SLAM and the EKF in observer-based SLAM. The first difference is the absence of an odometry update in observer-based SLAM. The observer’s position is a virtual position, since the observer does not actually move. The state transition model \( f \) in (4) now becomes only dependent on the previous state estimate:

\[ \hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}) \]  
(12)

In this case, \( f(\hat{\mathbf{x}}) \) is normally just a copying step, i.e.

\[ f(\hat{\mathbf{x}}) = \hat{\mathbf{x}} \]  
(13)

The second difference is the need to retrieve the absolute position of landmarks from the observer position and the local map features. In SLAM, the map features are absolute, i.e. stored in the global coordinate frame. Fig. 1 shows the virtual observer position and the landmarks that are measured relative to the observer. In observer-based SLAM, the state of each feature is defined in the local \( \mathbf{f}_1 \mathbf{f}_2 \) frame, and does not change from \( k = k_0 \) to \( k = k_1 \), where \( k_0 \) and \( k_1 \) are two arbitrary moments in time. The absolute positions of the landmarks are not directly available.
The estimated position of the landmarks is a transformation from the local $\mathcal{v}$-$\mathcal{w}$ frame to the absolute $\mathcal{v}$-$\mathcal{y}$ frame:

$$\hat{f}_{\text{abs},i} = R_\theta \cdot (f_i - \hat{\zeta}) \tag{14}$$

where $f_{\text{abs}}$ is the estimated absolute map of the environment, and $f_{\text{abs},i}$ is its $i$-th element. $R_\theta$ is the rotation matrix between the local $\mathcal{v}$-$\mathcal{w}$ frame and the absolute $\mathcal{v}$-$\mathcal{y}$ frame.

The observation model $h$ is a vector $h = [h_0, h_1, \ldots, h_n]^T$ with $n$ the number of landmarks, where each element $h_i$ is defined as:

$$h_i(x_{k|k-1}) = R_{\theta_{k|k-1}} \cdot (\hat{f}_{i|k|k-1} - \hat{\zeta}_{k|k-1}) \tag{15}$$

The innovation of measurement $i$ is then given by:

$$\hat{y}_{i,k} = z_{i,k} - R_{\theta_{k|k-1}} \cdot (\hat{f}_{i|k|k-1} - \hat{\zeta}_{k|k-1}) \tag{16}$$

From (10) and (13) it follows that the state transition matrix $F_{k-1}$ is a unity matrix. The observation matrix $H_k$ for measurement $i$ is defined by:

$$H_{i,k} = \frac{\partial h_i}{\partial x \mid_{x_{k|k-1}}} = \left[ \frac{\partial h_i}{\partial \zeta \mid_{x_{k|k-1}}} \frac{\partial h_i}{\partial \theta \mid_{x_{k|k-1}}} \frac{\partial h_i}{\partial f \mid_{x_{k|k-1}}} \right]_{x_{k|k-1}} \tag{17}$$

with

$$\frac{\partial h_i}{\partial \zeta \mid_{x_{k|k-1}}} = -R_{\theta_{k|k-1}} \tag{18a}$$

$$\frac{\partial h_i}{\partial \theta \mid_{x_{k|k-1}}} = \frac{\partial R_{\theta_{k|k-1}}}{\partial \theta_{k|k-1}} \cdot (f_i - \hat{\zeta}) \tag{18b}$$

$$\frac{\partial h_i}{\partial f \mid_{x_{k|k-1}}} = R_{\theta_{k|k-1}} \tag{18c}$$

The Kalman gain and the state and the covariance update are then calculated as follows:

**Algorithm II.1: State Update**

for $i \leftarrow 0$ to $n$

\[
g \leftarrow z_i - R_x \cdot (f_i - \hat{\zeta}_i) \\
H_x \leftarrow [-R_x; dR_x/d\theta(f_i - \hat{\zeta}_i); 0_{n \times d}]
\]

do

\[
S \leftarrow H_x P H_x^T + H_w W H_w^T \\
K \leftarrow P H_x S^{-1} \\
x \leftarrow x + K y \\
P \leftarrow (I - K H_x) P
\]

for $i \leftarrow 0$ to $n$

\[f_{\text{abs},i} \leftarrow R_\theta \cdot (f_i - \hat{\zeta}_i)\]

The proposed changes to the SLAM algorithm result in an accurate mapping method for moving objects viewed by a stationary observer. When the object is only partially visible, not all features will be updated upon a new measurement. However rigid body movement is extracted from the visible features, resulting in a shift in observer position. This shift also results in a shift in the absolute estimate of the invisible features, providing an accurate estimate of the object’s position and orientation.

### III. Modeled Landmark Interconnections

**Standard EKF SLAM** algorithms implicitly assume rigid feature interconnections. In general, estimation accuracy of the features estimated by these algorithms is improved by decreasing the covariance $Q$ of the process noise $w$. This results in a low predicted covariance estimate (8), resulting in a small Kalman gain (7). However, for non-rigid objects, where features move relatively, this is working counterproductive. This can be accounted for by implementing a dynamic model for the surface of the object, interconnecting all features dynamically.

**A. Extending the state transition of the EKF**

From equation 4 it follows that the state transition model $f$ in SLAM is just a copying step for map features. A model for connected map features can be included in this function $f$. This means that $f$ becomes:

$$f(x_{k-1}, u_k) = f \left( \left[ \hat{\zeta}_{k-1}; \hat{f}_{k-1} \right], u_k \right)$$

$$= f_u \left( \hat{\zeta}_{k-1}, \hat{f}_{k-1}, u_k \right) \cdot g \left( f_{k-1} \right) \tag{19}$$

with $f_u$ the odometry update and $g \left( f_{k-1} \right)$ the feature connection model. $f_u$ equals the first part of (4), so that $f_u$ equals $f(\hat{x}_{k-1}, u_k)$ without the feature list $f_{k-1}$. $g \left( f_{k-1} \right)$ describes the transition of the connections between the features in $f_{k-1}$. Hence $f_u$ and $g$ are independent and describe two different parts of the state vector $\hat{x}$. Note that $f(\hat{x}, u) \in C^1$, which means that it has to be derivable with respect to $\hat{x}$, i.e. $f(\hat{x}, u)$ and $g \left( f \right)$ are continuous.

This model can be extended in multiple ways, of which two will be discussed here. The first extension is the inclusion of previous state estimations:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, \hat{x}_{k-2|k-2}, \ldots, \hat{x}_{0|0}) \tag{20}$$
When we assume that these estimations are uncorrelated to \( \hat{x}_{k-1} \), the inputs to function \( f \) do not change, since \( \hat{x}_{k-1} \in \mathbb{N}^+ \) can be seen as parameters in \( f \), and hence the integrity of the Kalman filter is not affected.

The second extension is model parameter estimation. When model parameters in \( f \) are unknown, the EKF can be extended with states that represent these parameters. This is called an augmented extended Kalman filter [5][10][11]. These parameters can for instance be uncertain physical properties of the environment that is modeled. By including these parameters in de Kalman filter, instead of guessing their value, the model becomes more robust against changes in the environment.

**B. Influence on state estimation**

Including a model can improve accuracy and decrease sensitivity to noise when the model resembles the to be measured process, but can also deteriorate the position estimate when the model mismatch is large. Though a model mismatch may worsen the estimation, it can never destabilize the Kalman filter. The state transition \( F_{k-1} \) from (10) is the derivative of \( f \) with respect to \( x \). The predicted covariance estimate \( P_{k|k-1} \) is calculated by pre- and post-multiplying the covariance estimate from iteration \((k-1)\) with \( F_{k-1} \) (see Eq. 8). A large prediction update results in a large derivative \( F_{k-1} \), which results in a large covariance estimate \( P_{k|k-1} \). The Kalman gain from (7) is proportional with \( P_{k|k-1} \), and will hence be larger. This means that, if the prediction was wrong, the state update in this direction will be larger as a result of a larger Kalman gain and innovation \( \hat{y}_k \).

**IV. MODEL OF THE EYE**

The observer-based SLAM algorithm including the modeled landmark interconnections is especially suitable for tracking of the retina during vitreoretinal surgery. The eye is a deformable object of which the inside is viewed via the pupil through a microscope. During surgery, the pressure in the eye changes due to an infusion cannula. In this section, an eye-pressure model is derived and implemented in the EKF. We assume a stereo-vision camera to be the observer and vein junctions on the retina to be the landmarks.

**A. Assumptions and model properties**

Assumptions on the physical properties of the eye are:

1) the eye is isotropic
2) built up of homogeneous tissue
3) and linearly elastic

The eye is modeled by a pressurized hollow sphere of linearly elastic material. Fig. 2a visualizes part of the pressurized sphere. The pressure \( p(t) \) inside the eye exerts a force on the eye tissue. \( F_p(t) \) denotes the resulting stress in the tissue (boundary conditions).

In the model, map features are interconnected with springs along the boundary of a spherical shape. This way, landmarks on opposite sides of the eye are not connected directly, but indirectly via neighboring landmarks. Since we assume that the tissue of the eye is isotropic and homogeneous all springs have equal stiffness. To compensate for incorrect high stiffness in densely connected areas, the stiffness of each connection is multiplied with the normalized surface area of the adjacent tissue. In this model we assume map features to be steady, i.e. not in motion, and each step is a solution to a steady state, after which all stresses are assumed to be zero again.

When the pressure \( p(t) \) changes, we assume that all connections are elongated with the same factor \( e(t) \), which is directly proportional to the pressure \( p(t) \). This leads to a simplified model visualized in Fig. 2b. Since the pressure, and hence the elongation, is unknown, the elongation is augmented to the state vector in the EKF. It is assumed that the elongation changes slowly, hence the model in \( f \) (2) is extended with \( \dot{e}(t) = 0 \). \( e(t) \) will now be estimated with the resulting augmented EKF.

The boundary conditions that follow from \( F_p(t) \) in Fig. 2a do not exist in the elongation model in Fig 2b. This is solved by introducing a small spring force to the previous state. This makes solving the equations in the model possible.

**B. Implementation**

To implement the model, connections must be defined. All map features are used as points in a mesh generated using Delaunay triangles. This does not connect all landmarks, but only landmarks that are next to each other, which reduces computation time. Using Delaunay triangles for the mesh, the connections are well spread across the tissue that is represented by the map features.

The algorithm is computationally linear in time with respect to the number of landmarks and is therefore well suited for an algorithm with an in advance unknown number of features. The generated mesh consists of a list of linked features, that can be represented by their indexes. Now a \( u \times 2 \) list of connections is created, \( u \) being the number of unique connections in the mesh.

**Example IV.1**. Fig. 3 shows 4 points in two-dimensional space connected by a mesh created using Delaunay triangles. The corresponding list of connections is:

\[
\begin{bmatrix}
1 & 1 & 2 & 2 & 3
2 & 4 & 3 & 4 & 4
\end{bmatrix}^T
\]
Using these unique connections, the (virtual) forces on all map features can be calculated in terms of the positions of the neighboring features (according to the list of connections). Each connection is assumed to have a section surface size of $A$ and stiffness $E$, but since both parameters are to be estimated, they can be combined into a single parameter $G = AE$. The strain of each connection is denoted by $\epsilon$ and is calculated by:

$$\epsilon = \frac{\Delta L}{L} = \frac{l - L}{L}$$

(22)

where $L$ is the original length of the connection, and $l$ the current length. The tension $\sigma$ in a connection is found by $\sigma = E \cdot \epsilon$. The force from one connection on a map feature is then calculated using $F = A \cdot \sigma$. The direction of this force is always in the direction of the connection, so the force between two features $i$ and $j$ is:

$$F_{i\rightarrow j} = A \cdot \sigma \cdot \frac{f_i - f_j}{\|f_i - f_j\|_2} = G \cdot \epsilon$$

(23)

The total force at time-step $k$ is found by summing the forces of all connections to a landmark $i$:

$$F_{i,k} = \sum_{j=1}^{u_i} \frac{(f_{i,k-1} - f_{u_{i,j-1},k}) - (f_{i,k} - f_{u_{i,j},k})}{\|f_{i,k} - f_{u_{i,j},k}\|_2}$$

(24)

$$= G \sum_{j=1}^{u_i} \frac{(f_{i,k-1} - f_{u_{i,j-1},k}) - (f_{i,k} - f_{u_{i,j},k})}{\|f_{i,k} - f_{u_{i,j},k}\|_2}$$

where $u_{i,j}$ denotes the number of the $j$-th feature that is connected to feature $i$.

To minimize the stresses (i.e. forces) the following cost function is defined:

$$F_{\text{cost}}(f) = \text{tr}(F_i^T F_i)$$

(25)

Including the elongation $\epsilon$ equals solving the same stress balance equation, but with the previous connections elongated. Hence (24) becomes:

$$F_{i,k} = G \sum_{j=1}^{u_i} \frac{\epsilon \cdot (f_{i,k-1} - f_{u_{i,j-1},k}) - (f_{i,k} - f_{u_{i,j},k})}{\|f_{i,k} - f_{u_{i,j},k}\|_2}$$

(26)

As explained, the cost function only depends on differences between map features and hence the solution is singular. A second virtual force is introduced to “pull” the landmarks to their last position. This is done via a virtual spring with stiffness $K$:

$$F_{(i,k)\rightarrow (i,k-1)} = K(f_{i,k-1} - f_{i,k})$$

(27)

The total cost-function now becomes:

$$F_{\text{cost}}(f) = \text{tr}((f_{i,k-1} - f_{i,k})^T K f_{i,k-1} - f_{i,k}))$$

$$+ \text{tr}(F_i^T F_i)$$

(28)

which is non-linear, and can be minimized using optimization tools like the Mathworks Optimization Toolbox.

C. Polynomial approximation

Minimizing the nonlinear cost-function from (28) is computationally demanding and is therefore not preferred in a real-time application. If we assume that displacement of the landmarks is only small, (28) can be approximated using a polynomial. Eq. 26 becomes:

$$F_{i,k} = G \sum_{j=1}^{u_i} \frac{\epsilon \cdot (f_{i,k-1} - f_{i,k})}{\|f_{i,k} - f_{u_{i,j},k}\|_2}$$

(29)

where $f_{u_{i,j-1},k} = -c \cdot f_{u_{i,j-1},k-1}$ is assumed to be zero for landmark $i$. Eq. 27 is already linear and does not need to be altered. Hence the cost function (28) remains the same with a different definition for $F_i(f)$.

D. Solving the quadratic problem

Using the new definition for $F_i(f)$, the cost function in (28) becomes quadratic and hence has one global minimum. For finding this minimum we need to find a matrix multiplication form for (29), so the solution becomes trivial. The $u \times 2$ list of indexes is used for creating a $(2u) \times (n \cdot d)$ matrix $c$, such that $c \cdot f$ is a column vector containing the lengths of all connections, parted in frame coordinates (i.e. $x$ and $y$ in two dimensions). A block-diagonal matrix $N = \frac{1}{\|L\|_2}$ is created such that

$$N_i = \frac{(c \cdot f_{k-1} \cdot e)^T (c \cdot f_{k-1} \cdot e)}{\text{tr}((c \cdot f_{k-1} \cdot e)^T (c \cdot f_{k-1} \cdot e))^{3/2}}$$

(30)

We can now calculate the strain as a linear function of $f$:

$$\epsilon = A f + b$$

(31)

with

$$A = -c^T N c$$

(32)

$$b = -A f_{k-1} \cdot e$$

(33)

with $f$ a column vector of length $n \cdot d$, alternating with the coordinates.

Rewriting the cost-function (28) gives:

$$F_{\text{cost}}(f) = \text{tr}((f_{i,k-1} - f_{i,k})^T K (f_{i,k-1} - f_{i,k}))$$

$$+ \text{tr}(A f_{i,k} + b)^T G (A f_{i,k} + b)$$

(34)
which has a global minimum where \( \frac{\partial F_{\text{cost}}}{\partial f_{i,k}} \) equals zero, i.e. the vector \( f \) that minimizes (34) equals:

\[
f = -(GA + K)^{-1}(b - Kf_{i,k}) \tag{35}
\]

This is also the new definition of \( g(f_{k-1}) \) from (19). The state transition matrix from (10) becomes:

\[
F_{k-1} = (GA + K)^{-1}K \tag{36}
\]

**Example IV.2.** Using the list of connections from (21) we obtain:

\[c = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}
\]

which is a \( 10 \times 8 \) matrix with corresponding \( f \)

\[
f = \begin{bmatrix} 2.8 & 4.0 & 2.7 & 0.7 & 0.7 & 0.1 & 0.5 & 2.2 \end{bmatrix}^T \tag{37}
\]

a column vector of length 8, alternating with \( x \) and \( y \) coordinates.

**E. Augmented EKF**

If we now add the elongation \( e \) to the state vector in the EKF, the state transition matrix from (36) is extended with the derivative of model \( f \) w.r.t. \( e \):

\[
\frac{\partial f(x)}{\partial e} = 0 \quad \frac{\partial f(e)}{\partial e} = 1 \tag{38}
\]

\[
\frac{\partial f(f)}{\partial e} = (GA + K)^{-1}(A + \frac{\partial A}{\partial e}f \cdot e)
\]

**V. Simulation results**

In this section, the working and accuracy of the observer-based SLAM algorithm with the modeled landmark interconnections is demonstrated in simulations. The algorithm is implemented in \textit{Matlab} and two simulated experiments are conducted. The first experiment demonstrates the algorithm’s ability to accurately track temporarily invisible landmarks, the second demonstrates the algorithm’s ability to estimate the eye pressure. The input for both algorithms is equal, and consists of a two-dimensional point-cloud of random points with some noise added, simulating noisy position measurements. A comparison is made between the observer-based SLAM algorithm and standard Kalman filtering. The standard Kalman filter is a result of the extended Kalman filter without decoupling of relative movement of the map features from group movement, and without a model for the eye. The resulting filter is linear and has no coherence between its estimated states. Hence multiple standard Kalman filters with a single feature each are used. The Kalman filters are the basis for observer-based SLAM with modeled landmark interconnections and are hence well suited for comparison.

**A. Observer-based SLAM**

First, the algorithm’s ability to accurately track temporarily occluded landmarks is demonstrated. The input is rotated and translated over time. Fig. 4 shows the point-cloud and the results of the simulations of both mapping algorithms. At \( t = 0 \), both mapping algorithms create the same estimate of the environment. At \( t_1 = 0.33 \text{s} \), only the first half of the point-cloud is given as input, i.e. both algorithms only receive a position update of half the points in the point-cloud, simulating obstructed vision of the retina. At \( t_2 = 0.5 \text{s} \), the input has transformed to its final state and stops moving. The transformation is visualized in Fig. 4b. The simulation stops at \( t = 1 \text{s} \). The estimates of both methods at \( t = 1 \text{s} \) are shown in Fig. 4c. Due to the interconnections, the observer-based SLAM estimate overlaps with the input point-cloud. The estimate of the Kalman filters is only accurate up to \( t_1 \).
Fig. 5: Error plot from $t = 0$ to $t = 1$, observer-based SLAM (black) and separate Kalman filters (gray). The displayed error is the mean distance between each landmark estimate and the real position of the corresponding landmarks.

Fig. 5 shows the mean distance between all feature estimates and their corresponding landmarks, for the observer-based SLAM and the standard Kalman simulations. Before $t_1$ the complete input is still measured. In the observer-based SLAM algorithm (as well as in standard SLAM) the map features are connected, and multiple feature updates contribute to the estimate of one filter. The error of the observer-based SLAM algorithm is therefore smaller than the error of the separate Kalman filters.

At $t_1$ half of the landmarks are not measured anymore, but are still moving. The error of the Kalman filters then increases until the input stops moving ($t_2$), because half of its features are not updated anymore, while the corresponding features in the input point-cloud move away. When the movement stops, the error remains constant. The error of the observer-based SLAM algorithm increases slightly at $t_1$, since less measurements contribute to the estimate of the filter.

At $t_2$, the error of the landmarks that are still measured is approximately the same for both methods. When the number of points increases, more measurements contribute to the observer-based SLAM estimate, and the error decreases compared to separate Kalman filters.

**B. Modeled landmark interconnections**

Second, the algorithm’s ability to estimate the eye pressure is demonstrated. The input point-cloud does not rotate or translate, but the landmarks in the input move relative to each other. In discrete time, every iteration the distance between the landmarks in the input increases with a factor $e$. N.B. the resulting point-cloud after elongation does not depend on the mesh between the landmarks. In the simulation, the augmented EKF estimates the elongation from the position measurements, i.e. from the noisy input point-cloud. The prescribed elongation is denoted by $e$ and the estimated elongation is denoted by $\hat{e}$.

The eye model assumes a constant elongation $e$, i.e. $\dot{e} = 0$. To demonstrate robustness against model mismatches, the simulation does not. The prescribed elongation is shown in the top of Fig. 6. From $t = 0$ to $t = 0.5$, the elongation varies slowly (i.e. $\dot{e} \approx 0$). At $t = 0.5$, the elongation changes instantly, creating a large model mismatch ($\dot{e} \gg 0$). After $t = 0.5$, the elongation remains constant at 1 ($\dot{e} = 0$), meaning the landmarks in the input do not move relative to each other.

The EKF implementation is currently tuned such that it does not allow for large changes in $e$ ($\dot{e} \approx 0$). It can be tuned by changing the covariance of the elongation parameter estimate. A large covariance allows for a larger model mismatch, and the error will hence be lower at $t = 0.5$. However, it also causes the error to be larger when the model mismatch is small, because the elongation estimate will be more noisy.

The separate Kalman filters are currently tuned such that they do not allow for fast movements. This means that the estimate is good when the input point-cloud is approximately stationary. The estimate will be bad when the point-cloud expands fast due to the elongation of the connections. Tuning the filters such that the estimate is better for non-stationary input point-clouds is done by decreasing the covariance $W_k$ of the observation noise. This results in a smaller covariance of the innovation $S_k$ (9) and hence in a larger Kalman gain $K_k$ (7). This however results in more noise and hence larger errors in stationary situations, since the estimate changes more due to the increased Kalman gain.

The results of the simulation are shown in Fig. 6. At $0 \leq t < 0.5$ and at $0.5 < t \leq 1$, $e$ changes slowly, and hence the model mismatch in terms of the elongation is small, i.e. the maximum absolute difference between the simulated elongation speed and the modeled elongation speed is 0.033 mm/s. The estimate $\hat{e}$ of the elongation is therefore good with an average deviation of 0.01%. The rms value of the error of the EKF with eye model is $2.28 \cdot 10^{-4}$ mm. When the model mismatch is large ($t = 0.5$), the elongation estimate is bad with a deviation of 1%. The error at that point is 0.0034 mm. The mismatch is then resolved by the Kalman filter.

In the EKF implementation, multiple feature updates contribute to the same estimate, and hence the error is smaller than the error of the separate Kalman filters. Except at $t = 0.5$, where the model mismatch in elongation is large. From $t = 0$ to $t = 0.5$, the landmarks are moving due to the prescribed elongation, and hence the error of the separate Kalman filters is large, with an rms value of the error of 0.0020 mm, which is 9 times larger than the rms value of the error of the EKF implementation. From $t = 0.5$ to $t = 1$, the landmarks are stationary, and hence the error of the Kalman filters is constant with an rms value of 9.74 $\cdot$ 10$^{-4}$ mm. However, this is still more than 4 times larger than the error of the EKF with eye model.

The errors mentioned here are specific for these simulations, and will hence be different for different input point-clouds. In situations with larger model mismatches or less group movement of the input, the difference in error will decrease between the two methods. In situations with no model mismatch, larger elongations or more group movement, the difference in error will increase and observer-based SLAM with modeled landmark interconnections will perform better.
Fig. 6: Top: the prescribed elongation $e$ (dashed gray) and the estimated elongation $\hat{e}$ (black) are shown. Bottom: the average error per feature is shown for the EKF with the eye model (black) and separate Kalman filters (gray).

VI. CONCLUSIONS AND RECOMMENDATIONS

In this paper, a mapping method has been developed for a stationary sensor in a dynamical environment, which resembles an eye during surgery, viewed from a microscope. The method is an adaptation of Simultaneous Localization And Mapping (SLAM) and is called observer-based SLAM. It creates a map of the retina, given a set of vein junctions as input. By decoupling relative movement of the map features from group movement, higher accuracy can be achieved, and an estimate can be provided for map features that have not updated recently. A model exploiting a-priori knowledge of the physical interconnections between map features is included to further improve accuracy. Therefore, a model of the eye during surgery is derived that estimates expansion. Simulations demonstrate the accuracy improvement over standard filters, when the eye moves. In stationary situations there is no difference between observer-based SLAM and standard filters. When the eye moves or when vision is obstructed the improvement in estimation accuracy is significant. The mapping method can be tuned to be suitable for application under multiple situations, like fast movement or rapid changes in eye pressure. When the model does not resemble reality, i.e., when the properties of the eye differ from the assumed properties in the model, tuning can still allow for accurate mapping. Though accuracy will decrease when the model mismatch increases, with the right tuning the accuracy will never be worse than that of separate filters.

The current eye model solves a stress equation to a steady state and assumes all stresses to be zero again. This also enables estimation of external forces acting on the eye.

The mapping method’s robustness against failure in measurement association should also be improved. When measurement association fails, the filter in the mapping method can become unstable. Research should be done to find other filters that are less sensitive to association failure. Furthermore, methods should be searched for cleaning the map from invalid features, i.e., features that are not in the real world.

REFERENCES

Chapter 4

Implementation and experimental results

In this chapter the results of experiments are presented, demonstrating that observer-based SLAM with the modeled landmark interconnections is more accurate than separate Kalman filters. The results are validated via a ground truth and a comparison is made with a standard approach, i.e., a separate Kalman filter for each landmark.

Two separate experiments are conducted. The first experiment demonstrates the working and the accuracy of the observer-based SLAM part, i.e., accurate tracking of dots that are temporarily not visible. The second experiment demonstrates the working of the modeled landmark interconnections, i.e., the ability to accurately estimate the expansion of the eye and use this estimate to improve the estimate of the state of the vein junctions.

Both experiments use the same C++ implementation that is developed specifically for this application and for optimal performance of both observer-based SLAM using modeled landmark interconnections and separate Kalman filters. The algorithms and equations needed for implementation that are missing from the paper from Chapter 3 are presented in Appendix A.

In this chapter, observer-based SLAM using modeled landmark interconnections is referred to as algorithm 1, the separate Kalman filters are referred to as algorithm 2.

4.1 Experimental setup

The eye is approximated using a large rubber balloon. The physical properties of the balloon are assumed to be equal to the properties of an eye. Hence the model derived in section 3.IV is used, and the balloon is assumed to be:

1. isotropic
2. built-up of homogeneous material
3. linearly elastic.

The pressure inside the balloon is regulated using a faucet and a pressurized air stream, simulating the infusion cannula in an eye. The vein junctions are simulated by blue dots on the balloon. This simplifies the vision system needed for implementation.

The balloon is visualized by a Kinect sensor. Kinect is a line of motion sensing input devices by Microsoft for Xbox and Windows PCs. Kinect builds on range camera technology by using an infrared projector and camera. This 3D scanner system employs a variant of image-based 3D reconstruction. The sensor hence provides the user with a color image and a depth image of the environment. Metric 3D positions from points in the color image can then be extracted from these images.

Since Kinect is no ordinary video sensor, a software interface layer is made to connect it to the computer. The color image and the depth image are requested at maximum rate and medium resolution, that is $640 \times 480$ pixels at 30Hz. Requesting images at maximum resolution results in longer processing times, longer request times and hence a lower rate of 15Hz. When the images have been
received, they are processed to extract all blue pixels within a certain color range. The resulting black
and white image is filtered multiple times to reduce noise. A region growing algorithm is used to
detect blobs of pixels that represent the blue dots on the balloon. The 3D position of the center of each
blob is calculated by combining the 2D position in the color image with the depth image. To reduce
the risk for false positives, the detected dots outside a certain region in y-direction (axial direction of
the camera, i.e., towards the balloon) are rejected. The remaining set of dots are used as input to the
mapping and tracking algorithm.

A photo of the complete experimental setup is shown in Figure 4.1.

Figure 4.1: Experimental setup. In the bottom left the Kinect sensor, placed on a camera stand.
On the right the balloon with blue post-its glued on, hung from a stand. On the top, the balloon
is connected to a pipe with on the end a faucet and a pressurized air stream.

4.2 Ground truth

To be able to comment on the quality of the algorithm, a ground truth is established. Accurately
measuring the exact 3D positions of the blue dots is difficult, and hence the distances between the
dots are measured along the boundary of the balloon using a string. The dot pattern on the balloon is
shown in Figure 4.2.

4.3 Results

4.3.1 Experiment 1: tracking of temporarily invisible points

The first experiment that is conducted demonstrates algorithm 1’s ability to accurately track features
on the retina that are temporarily invisible. The balloon is used with the dots as in Figure 4.2. During
the experiment, the balloon is translated and rotated. The pressure inside the balloon will not change
during the experiment, since the faucet is closed. 14 seconds after the experiment has started, dot
number 3 is covered, while the balloon keeps translating and rotating. 28 seconds later, dot number
3 is uncovered again. After the balloon has stopped moving, some fast rotations and translations are
tested. Due to the connections between the points, temporarily invisible points are tracked accurately.

In Figure 4.3 the final estimate of the landmark positions of algorithm 1 and 2 are shown. Algo-
rithm 1 tracks all dots accurately. Dot number 3 is continuously tracked accurately, even when it is not
visible. Algorithm 2 loses track of dot 3 when it is covered. When it is uncovered at a different position,
algorithm 1 updates the position estimate of dot 3. Algorithm 2 fails to associate the new dot with dot 3
and adds a new Kalman filter to track the newly found dot. This is the grey cross at (150,170) in Figure
4.3.
Figure 4.2: 2D representation of the dots on the balloon, measured using the Kinect. The y-direction is in the paper and is not visualized here. Differences in y are small. The connections are created using Delaunay Triangles.

Fast movements are also tracked accurately by algorithm 1. All position updates of the dots contribute to the update of group movement, i.e. movement of the state $x$. Since fast movements are tracked, measurement association does not fail, and hence no false dots are added. Algorithm 2 sometimes fails to associate measurements when dots are moving fast. In the experiment, this resulted in one association failure. Therefore, there is an extra dot at (100,-40).

Algorithm 2 can be tuned to either allow for faster movement of the dots, or for no movement of the dots. When allowing for faster movement of the dots, the steady state estimate worsens. When tuning for no movement, fast movement cannot be tracked. This results in larger errors, and when the movement is large, measurement association may fail. In algorithm 1 this problem is reduced, since fast group movement is tracked from information of multiple feature updates.

The second column of Table 4.1 shows the lengths of the connections between the features. These lengths are assumed to be the ground truth. The mean absolute difference between the ground truth and the lengths of the connections between the estimates from algorithm 1 is 4.41 mm. The mean absolute difference between the ground truth and the lengths of the connections between the estimates from algorithm 2 is 4.83 mm. Hence the estimate of algorithm 1 is better. It is difficult to give a quantitative measure for improvement of algorithm 1 over algorithm 2, since the measurement input from the Kinect does not exactly agree with the ground truth. Hence the error of both algorithms is larger than in case a more accurate sensor – with the same precision and measurement noise – was used.

With this experiment, the working of observer-based SLAM has been demonstrated and validated. The tracking of temporarily invisible points, and fast group movement is more accurate than separate Kalman filters for each point. Adding false features to the map also does not happen in algorithm 1, under the conditions of the experiment, as opposed to algorithm 2. Hence algorithm 1 is more reliable and stable and gives a better position estimate.

4.3.2 Experiment 2: estimating change in expansion

The second experiment that is conducted demonstrates algorithm 1’s ability to estimate the expansion of the eye. Again, the balloon is used with the dots as in Figure 4.2. A few seconds after the experiment has started, the balloon is slowly deflated with a constant outward airflow, simulating a change in pressure from the pressure cannula in an eye. After just over four minutes, the airflow is stopped and the experiment is terminated.
The distances between the dots are measured before and after the experiment, these measurements are assumed to be the ground truth. From the change in the distances, the expansion of the balloon is calculated. Table 4.1 shows the measured distances at \( t_0 \) (start of the experiment) and at \( t_1 \) (end of the experiment). The calculated expansion is also shown. The mean expansion is 0.906 with a standard deviation of 0.027.

<table>
<thead>
<tr>
<th>from→to</th>
<th>dist. at ( t_0 )</th>
<th>dist. at ( t_1 )</th>
<th>expansion</th>
<th>dist. est. 1</th>
<th>diff. w/ g.t.</th>
<th>dist. est. 2</th>
<th>diff. w/ g.t.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 → 2</td>
<td>227</td>
<td>197</td>
<td>0.868</td>
<td>198.7</td>
<td>1.7</td>
<td>198.6</td>
<td>1.6</td>
</tr>
<tr>
<td>1 → 3</td>
<td>186</td>
<td>174</td>
<td>0.935</td>
<td>178.0</td>
<td>4.1</td>
<td>177.7</td>
<td>3.7</td>
</tr>
<tr>
<td>2 → 6</td>
<td>262</td>
<td>232</td>
<td>0.885</td>
<td>237.7</td>
<td>5.7</td>
<td>237.6</td>
<td>5.6</td>
</tr>
<tr>
<td>2 → 4</td>
<td>250</td>
<td>219</td>
<td>0.876</td>
<td>227.4</td>
<td>8.4</td>
<td>227.8</td>
<td>8.8</td>
</tr>
<tr>
<td>1 → 4</td>
<td>207</td>
<td>191</td>
<td>0.923</td>
<td>205.6</td>
<td>14.6</td>
<td>205.5</td>
<td>14.5</td>
</tr>
<tr>
<td>3 → 4</td>
<td>247</td>
<td>229</td>
<td>0.927</td>
<td>230.6</td>
<td>1.6</td>
<td>230.6</td>
<td>1.6</td>
</tr>
<tr>
<td>3 → 5</td>
<td>145</td>
<td>138</td>
<td>0.952</td>
<td>140.9</td>
<td>2.9</td>
<td>140.8</td>
<td>2.8</td>
</tr>
<tr>
<td>4 → 6</td>
<td>174</td>
<td>152</td>
<td>0.874</td>
<td>148.6</td>
<td>3.4</td>
<td>148.6</td>
<td>3.4</td>
</tr>
<tr>
<td>4 → 5</td>
<td>200</td>
<td>184</td>
<td>0.920</td>
<td>183.5</td>
<td>0.5</td>
<td>183.6</td>
<td>0.4</td>
</tr>
<tr>
<td>6 → 7</td>
<td>282</td>
<td>247</td>
<td>0.876</td>
<td>247.4</td>
<td>0.4</td>
<td>248.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4 → 7</td>
<td>228</td>
<td>208</td>
<td>0.912</td>
<td>207.7</td>
<td>0.3</td>
<td>207.4</td>
<td>0.6</td>
</tr>
<tr>
<td>5 → 7</td>
<td>168</td>
<td>155</td>
<td>0.923</td>
<td>154.2</td>
<td>0.8</td>
<td>154.2</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Mean: 0.906 | Mean: 3.70 | Mean: 3.73

| SD: 0.027 |

Table 4.1: Experimental results of expansion estimation. The first column shows the indexes of the points that build up the connection. The second and third column show the distances in mm between the points in the first column at the start \( (t_0) \) and end \( (t_1) \) of the experiment. The fourth column shows the calculated expansion of the connection. Distance estimates 1 and 2 are the estimates of observer-based SLAM and separate Kalman filters respectively. “diff. w/ g.t.” denotes the difference with the ground truth (the measured distances in the third column). N.B. column five does not contain the results of the first experiment in this subsection. It contains the results of the second experiment, better tuned for a model mismatch.

The estimated expansion in the experiment is at each iteration relative to the previous iteration. The final estimated expansion of the balloon is then found by multiplying all relative expansions. The
result is an estimated expansion of 0.909, which is very close to the calculated expansion from the measurements, given the large standard deviation.

The large differences in the calculated expansions also shows that the balloon is not built up of homogeneous material. This causes a mismatch between the relative movement of the points on the balloon and the model, when the pressure inside the balloon changes. Due to this mismatch, the position estimation of the dots in algorithm 1 is worse than the position estimation of the dots in algorithm 2. The mean difference in connection lengths between the estimate of the filters and the measurement is 4.80 mm for algorithm 1 and 3.73 mm for algorithm 2.

To improve the estimate of the dots, the algorithm must allow for more relative movement. To allow for more relative movement that is not captured with the model, some model and filter parameters can be tuned. The tissue stiffness in the model is decreased, and the covariance of the process noise ($Q$) of the map features is increased. This only influences relative movement. The changes result in the estimates shown in the fifth column of Table 4.1. The mean difference in connection lengths between the estimate of algorithm 1 and the measurement is now 3.70 mm, which is approximately equal to the mean difference in connection lengths between the estimate of algorithm 2 and the measurement (3.73 mm). Since algorithm 1 now allows for relative movement that is inconsistent with the model, the expansion estimate is now incorrect.

In the simulations in Section 3.V the model mismatch in terms of inconsistent relative movement is smaller than in the experiments conducted with the balloon. So when comparing the experimental results to simulation results, we see that, when the model mismatch is smaller, the position estimates of algorithm 1 will be better than those of algorithm 2. When the model mismatch is large, the estimate of algorithm 2 is better than the estimate of algorithm 1. We conclude that, with the right tuning, the position estimates of algorithm 1 will never be worse than those of algorithm 2.

### 4.4 Conclusions

The conducted experiments demonstrate and validate observer-based SLAM. The algorithm is able to accurately track temporary occluded landmarks, while separate Kalman filters are not. Due to multiple measurement updates for the same filter, group movement is tracked and algorithm 1 can follow fast movements better than algorithm 2. Due to the ability to follow fast movements, measurement association does not fail quickly. The overall estimate of the features is better in case of observer-based SLAM, compared to separate Kalman filters.

Accuracy improvement due to the inclusion of a model in the landmark interconnections has not been validated in the conducted experiments. The assumptions on the physical properties of the balloon where too different from the actual properties, resulting in a large model mismatch. In this case, the estimate of separate Kalman filters is better than the estimate of observer-based SLAM with modeled landmark interconnections. With the right tuning, the position estimate will however still be equally good as or better than the estimates of separate Kalman filters.
Chapter 5

Conclusions and recommendations

In this work, a method is derived for accurately mapping and tracking the retina during vitreoretinal surgery. An ideal visualization modality for intra-operative real-time retina tracking is stereo vision with OCT probe addition. The current surgical setting already contains a stereo vision microscope which can easily be extended with cameras for stereo vision, which is not expensive. The OCT probe can be implemented in the trocar which will provide extra safety. Stereo vision gives three-dimensional images of the retina which can be processed to extract vein junctions. These vein junctions serve as input to a mapping and tracking algorithm. Real-time retina mapping and tracking can be achieved using standard SLAM employed in an observer-based setting. This is called observer-based SLAM. It keeps track of temporary occluded points and allows for eye pressure estimation. The eye pressure estimation is used in a eye-pressure model of the eye that improves the accuracy of the map. Simulations and experiments show that the developed mapping and tracking algorithm performs better than standard Kalman filters in case of steady eye pressure and in case of an eye with properties that match the model properties. The algorithm performs equal to standard Kalman filters in case of a large mismatch in modeled eye properties.

The current version of the eye model is not suitable for application in the eye. The first step in improving the model is extending the model to include non-homogeneous tissue. As the experiments showed, the current model mismatch with respect to homogeneousness of the tissue cause incorrect landmark estimates. As the eye is also not perfectly homogeneous, non-homogeneous tissue is to be included. The current model does also not build up stress in tissue, and solves the stress equation to a steady state at each moment in time. A next version of the model should estimate the stresses – most probably with an augmented extended Kalman filter – and use these to provide a better estimate of the map features. This also allows to estimate external forces on the eye. After the model update, the selected visualization modality, with vein junction recognition implemented, and the observer-based SLAM with modeled landmark interconnections should be tested on a real eye.
References


Appendix A

Implementation of observer-based SLAM with modeled landmark interconnections

In this appendix, all information necessary for implementation of observer-based SLAM is provided. First, the initial map and feature list are discussed. Second, the map will be extended with new features. Then the equations for updating the observer state and feature list are given. And last, a possible observer update method will be discussed.

A.1 Initial map and feature list

The observer-based SLAM algorithm is based on the extended Kalman filter implementation. The state vector in this filter is defined as:

$$x = \begin{bmatrix} \zeta \\ \theta \\ e \\ f \end{bmatrix}$$  \hspace{1cm} (A.1)

with \( \zeta \in \mathbb{R}^{3\times1} \) and \( \theta \in \mathbb{R}^{3\times1} \) the translation and rotation of the observer respectively, \( e \) the relative expansion of the connections between map features, and \( f \) the feature list, currently empty. \( \zeta \) and \( \theta \) are both zero vectors and \( e \) is 1.

The covariances of the process noise \((Q)\) and observation noise \((W)\) are determined and if necessary tuned. The covariance of the state vector \((P)\) is initialized at:

$$P = \begin{bmatrix} Q_\zeta & Q_\theta \\ Q_\theta & Q_e \end{bmatrix}$$  \hspace{1cm} (A.2)

where \( Q_\zeta, Q_\theta \) and \( Q_e \) are the covariances of \( \zeta, \theta \) and \( e \) respectively.

A.2 Update map with new features

Due to erroneous feature detections caused for example by moving objects or measurement noise, additional care has to be taken to filter out interfering measurements. For any detected object that cannot be explained by existing features, a new feature candidate is generated but not put into the map directly. Instead it is added into a provisional list with a weight representing its probability of being a useful feature. In the next measurement step, the newly arrived candidates are checked against all candidates in the waiting list; reasonable matches increase the weight of corresponding candidates. Candidates that are not matched lose weight because they are more likely to be a moving object or measurement noise. When a candidate has its weight above a certain threshold, it is added to the map.
When a measurement \( z \) is added to the waiting list, the measurement is first translated and rotated to the local coordinate frame \( \vec{v} \vec{w} \):
\[
z_{\vec{v} \vec{w}} = R_{\theta}^{-1} \cdot z + \zeta
\] (A.3)

Adding the candidate feature to the map increases the feature list \( f \) with the position stored in the waiting list. The size of the feature list is then \( 3n \times 1 \), \( n \) being the number of features in the map. The covariance matrix \( P \) is extended as follows. Define the function \( g ([\zeta; \theta], z) \) that computes the absolute coordinates of the measurement \( z \). Define \( G_{\zeta, \theta} \) as the Jacobian of \( g \) with respect to the observer state:
\[
G_{\zeta, \theta} = \frac{\partial g ([\zeta; \theta], z)}{\partial [\zeta; \theta]} \tag{A.4}
\]
Define \( G_z \) as the Jacobian of \( g \) with respect to the measurement:
\[
G_z = \frac{\partial g ([\zeta; \theta], z)}{\partial z} \tag{A.5}
\]
Define \( M \) as:
\[
M = \begin{bmatrix}
I_{7+3n} & 0_{(7+3n) \times 3} \\
G_{\zeta, \theta} & 0_{3 \times 3n} & G_z
\end{bmatrix} \tag{A.6}
\]
The extended covariance matrix \( P_{ext} \) then becomes:
\[
P_{ext} = M \begin{bmatrix} P & W \end{bmatrix} M^\top \tag{A.7}
\]

In case of observer-based SLAM, the observer does not actually move, and hence \( z \) is already absolute. \( G_{\zeta, \theta} \) and \( G_z \) now simplify to a zero matrix and an identity matrix respectively. The calculation of \( P_{ext} \) simplifies to:
\[
P_{ext} = \begin{bmatrix} P & W \end{bmatrix} \tag{A.8}
\]

### A.3 Update existing features with new measurements

All new measurements are compared to the features in the feature list. Any new measurement is transformed to the local coordinate frame \( \vec{v} \vec{w} \), as in Equation A.3. Association can be based on two factors, the Mahalanobis distance \([20]\) or a tuned threshold. The Mahalanobis distance takes the covariance of the feature into account, which means that uncertain features are associated even when at a greater distance. A tuned threshold does not.

When a new measurement is associated with a feature already in the map, the position of this feature is updated as in section 3.II.A. The state transition model \( f \) from Equation 3.2 only depends on the feature list \( f \) and the extended state \( e \). This means that the state transition matrix \( F = \frac{\partial f}{\partial x} \bigg|_{\hat{x}} \) has the following layout:
\[
F = \begin{bmatrix}
\zeta & 0 & e & f \\
\dot{\zeta} & 0 & I_3 & I_3 \\
\dot{\theta} & 0 & I_3 & 1 \\
\dot{e} & 0 & I_3 & \frac{\partial f}{\partial e} \\
\end{bmatrix} \tag{A.9}
\]
The values of \( \frac{\partial f}{\partial c} \bigg|_{\hat{x}} \) and \( \frac{\partial f}{\partial f} \bigg|_{\hat{x}} \) are given in (3.40) and (3.36) respectively.

The covariance matrix \( P \) is updated as follows:
\[
P_k = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^\top + K_k W K_k^\top \tag{A.10}
\]

\( P_{k|k} \) can become asymmetric from numerical causes. To ensure its symmetry the following operation is performed:
\[
P_{k|k} = \frac{1}{2} \left( P_k + P_k^\top \right) \tag{A.11}
\]
A.4 Possible observer update: Iterative Closest Point

Since there is no incremental measurement that serves as initial guess for the movement of the environment, the odometry update is removed from observer-based SLAM. This however means that any discontinuities in sampling or any large translations or rotations can lead to failure in measurement association. A solution to this problem is calculating a transformation between the last state of the map and the new measurement. This transformation then serves as odometry update.

Iterative Closest Point (ICP) is an algorithm employed to minimize the difference between two point-clouds [21]. In the algorithm, one point-cloud, the target, is kept fixed, while the other one, the source, is transformed to best match the target. The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize the distance from the source to the reference point cloud. Essentially, the algorithm steps are:

1. For each point in the source point-cloud, find the closest point in the reference point cloud.
2. Estimate the combination of rotation and translation using a mean squared error cost function that will best align each source point to its match found in the previous step.
3. Transform the source points using the obtained transformation.
4. Iterate (re-associate the points, and so on).

When a successful transformation is found, the chances of failure to associate map features to the measurement are much smaller.

This method only works when the measurement point-cloud and the map have most points in common. So when many new landmarks are found in the measurement, or when vision is obstructed and only a few landmarks are re-observed, a successful transformation cannot be found.
Appendix B

Real-time three-dimensional object detection in vitreoretinal surgery
Results of a Bachelor Thesis

During this master thesis, a bachelor student has been researching real-time three-dimensional object detection in vitreoretinal surgery. The focus was on tool recognition and stereo-vision using a stereo-microscope. This appendix provides a short summary of the results. For an in-detail explanation please read the complete bachelor thesis [22].

B.1 Undistorting and rectifying camera images

Manufacturing errors of lenses may result in distortions of the acquired images. Radial and tangential distortion can be calibrated by taking images of a chessboard. The distances between the boxes is known, and all lines are straight and parallel or perpendicular. By recognizing the corners of the boxes on the chessboard, a model can be made of the distortion of the lens. Inverting this model undistorts the camera images.

B.2 Stereo-vision

Stereo-vision is based on comparing points in two camera images, that are recorded from slightly different angles. Figure B.1 shows a typical stereo-vision setup. The so-called disparity $d$ is a measure for the distance ($z$) to the point and is calculated as:

$$d = x_l - x_r = f \frac{b}{z} \tag{B.1}$$

The resolution of the disparity is limited by the smallest measurable distance between $x_l$ and $x_r$. It hence equals the resolution of the camera $r$. The resolution $z_r$ of the depth measurement is therefore:

$$z_r = f \frac{b}{r} \tag{B.2}$$

When $z_r$ is too large for the purpose, it can be decreased by decreasing $f$ or $b$, or by increasing $r$.

B.3 Tool recognition

Before detection, the image is blurred for reducing noise and camera artefacts. Although some details in the image might be removed, object detection tends to be more successful when applied on a blurred image.

Tool recognition is performed using one or more of the following methods:
Figure B.1: Typical stereo vision setup. A point \( P(x, y, z) \) in three-dimensional space is recorded from two cameras. The point is recorded in the orange dots, with pixel positions \( x_l \) and \( x_r \) for the left and right dot respectively. \( f \) is the focal distance of the camera lens.

1. Color detection
2. Background substraction
3. Edge and line detection
4. Template matching

Color detection is based on the fact that the tool always has a substantially different color than the rest of the eye. This is not always the case, e.g. bad lighting can create unwanted shadows that can result in false positives. Color detection is done in the HSV color space. The advantage of the HSV is that the color information is saved in one of the three channels, greatly simplifying of detecting a certain color.

Background substraction is based on the difference between a preoperative image of the retina, and intra-operative images. The preoperative image does not contain the tool, and hence the difference between the two images is the tool, extracted from the intra-operative image. This method only works when a clear image of the retina and a fixed viewing point are guaranteed.

Line detection is done using the Canny edge detection algorithm and the Hough transform. Canny edge detection detects edges in a gray-scale image, based on the intensity of neighboring pixels. After the edges in an image have been detected, straight lines most probably belong to the tool. The linear Hough transform is used for detecting straight lines.

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. The tool can be manually selected by the surgeon in an image of the retina with tool. The selected area is called the template image. During surgery, this template image is sought in live retina footage.

### B.4 Retina tracking

By dividing the image of the retina in small parts, and creating a template of each of these parts, the retina can be tracked using template matching. When parts of the retina are found that are not in any of the templates, a new template can be created. This way, translation and rotation of the retina can be tracked.