



## 3D STEREOVISION FOR QUANTIFICATION OF SKIN DISEASES

AN INITIAL STEP TOWARDS ACHIEVING  
SUB-MILLIMETRE 3D RECONSTRUCTION

N. (Nimish) Shah

MSC ASSIGNMENT

**Committee:**

dr.ir. Ferdi van der Heijden  
prof.dr.ir. Wiendelt Steenbergen  
dr. Françoise Siepel

November, 2020

049RAM2020  
Robotics and Mechatronics  
EEMathCS  
University of Twente  
P.O. Box 217  
7500 AE Enschede  
The Netherlands

UNIVERSITY  
OF TWENTE.

TECHMED  
CENTRE

UNIVERSITY  
OF TWENTE.

DIGITAL SOCIETY  
INSTITUTE

**Electronic version of this document can be availed  
at <https://essay.utwente.nl/85105/>.**  
For better visibility of figures in this document please  
refer to the electronic version.

*This page is intentionally left blank.*



Please consider your environmental responsibility. Save per page about 200 mL of water + 2 g of CO<sub>2</sub> + 2 g of wood. Print Duplex.





# Summary

Skin diseases are among the most common health problems worldwide and are associated with considerable comorbidities. The objective and accurate quantification of the severity is essential to monitor the progress of the disease and healing, observe the effects of medication, and effective decision-making on the treatment method. Psoriasis is one of such diseases affecting 2-3% of the global population. To evaluate the degree of severity of the disease PASI index is widely used. This method assigns the single objective score ranging from 0 to 72; nonetheless, each region's score is determined subjectively.

As part of the 3DHAP project, the purpose of the thesis is to design a stereo camera system capable of differentiating between induration levels of PASI. The stereo camera system has limited overlap, which means that only a limited area can be reconstructed. To increase the area of reconstruction error-state Kalman filter visual SLAM algorithm is investigated. Experiments are conducted on both simulated data and real data. In the case of simulated data, the use of a gyroscope is also investigated.

The initial prototype of the stereo camera has sufficient depth resolution to differentiate between the different levels of induration in accordance with PASI. The ES-EKF SLAM is also able to generalize the camera trajectory and reconstruct the object both using simulated and real data. Further, it is also seen that the ES-EKF SLAM with only visual information is not robust, and additional sensors are required.

## सारांश

त्वचा रोग दुनिया भर में सबसे आम स्वास्थ्य समस्याओं में से एक हैं और कई सहरुगणताओं से जुड़े हुए हैं। रोग की प्रगति और उपचार की निगरानी करने, दवा के प्रभाव का निरीक्षण करने और उपचार पद्धति पर प्रभावी निर्णय लेने के लिए बीमारी की गंभीरता का सटीक मात्राकरण आवश्यक है। सोरायसिस २ से ३% वैश्विक आबादी को प्रभावित करने वाली ऐसी बीमारियों में से एक है। इस बीमारी की गंभीरता की मात्रा का मूल्यांकन करने के लिए PASI सूचकांक का व्यापक रूप से उपयोग किया जाता है। यह विधि ० से ७२ तक की सीमा से एक पूरक अंक प्रदान करती है। परन्तु प्रत्येक क्षेत्र के अंक व्यक्तिपरक ढंग से निर्धारित किये जाते हैं।

3DHAP प्रकल्प के अंतर्गत इस शोध-प्रबन्ध का उद्देश्य एक स्टीरियो कैमरा प्रणाली की रचना करना है कि जो PASI के संकेत स्तरों के बीच अंतर करने में सक्षम हो। स्टीरियो कैमरा प्रणाली में सीमित अधिव्यापन होता है जिसकी वजह से केवल सीमित क्षेत्र का पुनर्निर्माण किया जा सकता है। पुनर्निर्माण के क्षेत्र को बढ़ाने के लिए Error-State Kalman Filter SLAM विधि की जांच की गयी। यह प्रयोग अनुरूपित डेटा और वास्तविक डेटा पर किए गए हैं। अनुरूपित डेटा के मामले में जाइरोस्कोप के उपयोग की भी जांच की गई है।

स्टीरियो कैमरा का यह प्रारंभिक नमूना PASI के अनुसार अलग-अलग गहराई स्तरों के बीच अंतर करने के लिए सक्षम है। ES-EKF SLAM कैमरा प्रक्षेपण का व्यापक अनुमान लगाने और अनुरूपित और वास्तविक डेटा दोनों का उपयोग कर वस्तु का पुनर्निर्माण करने में भी सक्षम है। इसके अलावा यह अवलोकित किया कि केवल दृश्य की जानकारी के साथ ES-EKF SLAM सुदृढ़ नहीं है और अतिरिक्त संकेतकों (सेंसर) की आवश्यकता है।



# Contents

<b>Summary</b>	<b>iii</b>
<b>Notations and Conventions</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	1
1.2 Related Work . . . . .	2
1.3 Organization of the Thesis . . . . .	4
<b>I Design of the Stereo System</b>	<b>7</b>
<b>2 Background</b>	<b>9</b>
2.1 Field of View . . . . .	9
2.2 Depth of Field . . . . .	9
2.3 Coordinate Systems and Transformations . . . . .	10
2.4 Camera Models . . . . .	11
2.5 Epipolar Geometry . . . . .	13
2.6 Depth Perception . . . . .	14
2.7 Calculation of Geometrical and Optical Requirements . . . . .	15
2.8 3D Landmark Estimation from Multiple Images . . . . .	16
<b>3 Design and Implementation</b>	<b>19</b>
3.1 Identification of the Geometric and Optical Parameters . . . . .	19
3.2 Calculation of the Stereo-geometry . . . . .	20
3.3 The Camera Geometry Visualiser App . . . . .	20
<b>4 Experiments</b>	<b>25</b>
4.1 Motivation and Goals . . . . .	25
4.2 Methods . . . . .	25
4.3 Results . . . . .	27
4.4 Discussions . . . . .	29
<b>II 3D Reconstruction using Visual SLAM</b>	<b>31</b>
<b>5 Background</b>	<b>33</b>
5.1 Error-State Extended Kalman Filter . . . . .	33
5.2 Simultaneous Localization and Mapping . . . . .	35
5.3 Observability Analysis . . . . .	38
<b>6 Design and Implementation</b>	<b>41</b>
6.1 Kinematics of ES-EKF SLAM . . . . .	41
6.2 Implementation . . . . .	43

<b>7 Experiments</b>	<b>49</b>
7.1 Motivation and Goals . . . . .	49
7.2 Methods . . . . .	49
7.3 Results . . . . .	51
7.4 Discussions . . . . .	53
<b>8 Conclusions</b>	<b>57</b>
<b>Acknowledgements</b>	<b>59</b>
<b>A Definitions</b>	<b>61</b>
A.1 Image Sensor Metrics . . . . .	61
A.2 Lens Metrics . . . . .	62
A.3 Stereo Geometry . . . . .	62
<b>B PASI and PASImeter</b>	<b>63</b>
<b>C Workpiece Drawing</b>	<b>65</b>
<b>References</b>	<b>67</b>

# Notations and Conventions

$\vec{x}$	a vector $x$
${}^a\vec{x}$	a vector $x$ in coordinate system $a$
${}^b\vec{t}_a$	a translation vector $t$ describing the rotation of coordinate system $b$ w.r.t. the coordinate system $a$
$\mathbf{R}$	a matrix $R$
${}^b\mathbf{R}_a$	a rotation matrix $R$ describing the rotation of coordinate system $b$ w.r.t. the coordinate system $a$
$\underline{\mathbf{H}}$	a homogeneous matrix $H$
$\underline{\vec{x}}$	a vector $x$ represented in homogeneous coordinate system
${}^b\mathbf{v}_{a,c}$	a quantity $v$ of the body $c$ relative to the $a$ and expressed in the coordinate system $b$

For the sake of brevity, the symbol  $\hat{\cdot}$  over the vectors in used in the SLAM part of this document is omitted. But for the sake of clarity between vector and scalars, both vectors and matrices are typeset in bold.

To be consistent all figures concerning length are represented in  $mm$ , if mentioned otherwise.



# List of Figures

1.1	Experimental Setup of Bloemen et. al. (2011) . . . . .	3
1.2	Output of PRIMOS 3D Optical Scanner . . . . .	3
1.3	Experimental Setup of Ahmad Fadzil et al. (2013) . . . . .	3
2.1	Field of View . . . . .	9
2.2	Pin-hole camera geometry . . . . .	12
2.3	The Geometry of Two Cameras . . . . .	14
2.4	Triangulation in a Stereo Vision system . . . . .	15
3.1	Depth Resolution Plots . . . . .	21
3.2	Camera Simulator App . . . . .	22
3.3	Stereo Camera Setup . . . . .	23
4.1	Workpiece used for Reconstruction . . . . .	26
4.2	Patterns Drawn on the Workpiece . . . . .	26
4.3	High Contrast Pattern for Auto-Focus . . . . .	27
4.4	Montage of Left and Right Undistorted Images . . . . .	28
4.5	Mesh of the Yellow Pattern . . . . .	28
4.6	Regions of Interest of the Mesh . . . . .	29
4.7	Height Map of the Reconstructed Workpiece . . . . .	29
6.1	Camera Calibration . . . . .	44
6.2	Simulated Data Collection . . . . .	44
6.3	Globe used for Real Data Collection . . . . .	45
6.4	Real Data . . . . .	45
6.5	Matched Features using KAZE Descriptors on the Real Data . . . . .	46
6.6	The Proposed SLAM Pipeline . . . . .	46
7.1	Results of the Experiments . . . . .	51
7.2	Trace of the Error-State Covariance Matrix . . . . .	52
7.3	3D Reconstruction of the Globe . . . . .	52
7.4	Correlation Plots . . . . .	52
A.1	Lens and Sensor Metrics . . . . .	61
C.1	Isometric View of the Validation Workpiece . . . . .	65



# List of Tables

3.1	Effect of Increasing Adjustable Parameters on the Performance Parameters . . . . .	19
4.1	Camera Calibration Results . . . . .	27
4.2	Reconstructed Heights . . . . .	28
7.1	Proposed Experiments . . . . .	50
7.2	Radius of the Fitted Sphere . . . . .	53
B.1	PASI Explanation . . . . .	63



# Chapter 1

## Introduction

Skin diseases are among the most common health problems worldwide and are associated with a considerable burden. The burden of skin disease is a multidimensional concept that encompasses the psychological, social, and financial consequences of skin disease on the patients, their families, and society. It pervades all cultures, occurs at all ages, and affects between 30% and 70% of individuals (Hay et al., 2014). Skin is frequently damaged because it is directly in the *firing line*, and for this reason, skin diseases are very common. There are more than 3000 known diseases of the skin (Basra & Shahrukh, 2009).

This thesis focuses on *psoriasis*, a chronic, non-communicable, painful, disfiguring, and disabling disease with no cure affecting about 2–3% of the world population (Hurriyatul, Fadzil, & Hani, 2010; World Health Organization, 2016). The cause of the disease is the immune system's failure to signal the skin to grow new skin cells correctly. Instead, this condition accelerates the growth cycle of skin cells from the normal 28 days to 4 days (Williams & Wilkins, 2005).

Psoriasis involves the skin and nail and is also associated with various co-morbidities. Psoriasis skin lesions are localized or generalized, mostly symmetrical, sharply demarcated, red papules and plaques, and usually covered with white or silver scales (World Health Organization, 2016). The lesions get thicker with coarse white scales at higher severity levels. Lesions cause itching, stinging, and pain. Although psoriasis is incurable, it is a treatable skin disorder. The objective and accurate quantification of the severity is essential to monitor the disease's progress and healing, observe the effects of medication, and effective decision-making on the treatment method.

The *psoriasis area and severity index* (PASI) score is used to evaluate the degree of clinical manifestations of psoriasis objectively (Fleischer et al., 1996). PASI is a gold standard to assess psoriasis severity as well as the treatment efficacy. The calculation of PASI is discussed in Appendix B.

### 1.1 Problem Statement

Although a gold standard, the determination of PASI is very tedious and complex. Hence, in practice, the PASI score is determined subjectively by dermatologists using their visual and tactile senses; therefore, direct contact between doctor and patient is required. This subjective scoring results in inter- and intra-observer variability (Hurriyatul et al., 2010; Fadzil et al., 2010). Furthermore, as mentioned by Langley and Ellis (2004), PASI assessment gets more reliable with experience.

This assignment is a part of the project 3DHAP, which aims to develop a compact handheld device which can cope with the inevitable movements between the instrument and the subject, and scan a large tissue area by manually *sweeping* the device over the tissue in a controlled but flexible manner. The ultimate goal of this project is to stitch such perfusion information together and make a make 3D and AR visualization for showing the perfusion information. Deriving from this, the goal of this assignment is *to design a stere-*

*ocamera system which is able to differentiate between different induration levels of PASI, while increasing the area of reconstruction.*

In the case of psoriasis, the *induration* of the diseased site can be measured. The scale for classifying *induration* should be in accordance with the PASImeter (see Appendix B) developed by biopharmaceutical company AbbVie Deutschland GmbH & Co. KG. This scale has five induration classes (0–4), each varied by 0.25 mm (Prof. Dr. med. Kristian Reich, 2013).

The end goal is divided into the following sub-goals:

1. finding the geometrical and optical requirements for a stereo camera setup for stereo-photogrammetry to obtain the required depth resolution.
2. to increase area captured, investigate the initial possibility of ES-EKF SLAM by means of simulations and on real data.

Due to country-wide COVID-19 restrictions fulfilment of these goals will be evaluated on the workpiece with specifications similar to those found in Psoriasis patients.

## 1.2 Related Work

Computer vision technique has shown great application in surgery and therapy of some diseases. Recently, 2D-3D image analysis, 3D modelling and rapid prototyping technologies have driven the development of medical imaging modalities (Gao, Yang, Lin, & Park, 2018). 2D and 3D image analysis techniques are also used in quantification, treatment and monitoring of psoriasis. Most studies in this field have a focus on the assessment of *erythema*; which is carried out by 2D images. Further, few researchers have taken advantage of 3D computer vision techniques to assess the severity of the *induration* factor. This section briefly summarizes the various research done in the area of objective quantification of the severity of psoriasis and the state-of-the-art visual SLAM implementations for anatomical 3D reconstruction using handheld device.

### 1.2.1 Measurement of 3D relief

Bloemen, van Gerven, van der Wal, Verhaegen, and Middelkoop (2011) and Jacobi et al. (2004) use Phase-shift Rapid In Vivo Measurement of the Skin (PRIMOS), an instrument developed by GFMesstechnik GmbH, Teltow, Germany to measure the surface roughness of skin and scars. This measuring device produces a 3-dimensional image of the skin using digital stripe projection technique. A chip absorbs the reflected light with a high-resolution camera and the height differences in the measured area are recorded. The fig. 1.1 shows the experimental setup. The field of view of the PRIMOS is 30 mm × 40 mm. Bloemen et al. (2011) report intraclass correlation coefficient of 0.85 for skin and scar classes was obtained using proprietary PRIMOS software. Fadzil et al. (2010) and Ahmad Fadzil et al. (2013) also use this instrument to measure the surface roughness of psoriasis lesion. The lesion is modelled by a rough surface, and roughness is determined using polynomial surface fitting. fig. 1.2 shows the output of PRIMOS 3D optical scanner. Similar techniques are also used in Hani and Prakasa (2014). Park et al. (2004) evaluated the feasibility of PASI-scoring system using two CCD cameras (TM4401®, Pulnix Co, CA, USA) and concluded that the stereo-vision could be considerably useful in overcoming the difficulties in the objective and quantitative result interpretation for evaluating the conditions of the skin contour in the different cutaneous disorders. Hurriyatul et al. (2010) obtain acquisition accuracy of 0.1 mm Konica Minolta Non-Contact 3D Digitizer VIVID 910 laser scanner. A 5 step algorithm is developed to measure lesion thickness. Initially, a 3D image is captured, then it is segmented from healthy skin. In the next steps, lesion thickness is calculated by constructing a lesion base, and appropriate PASI score is assigned.

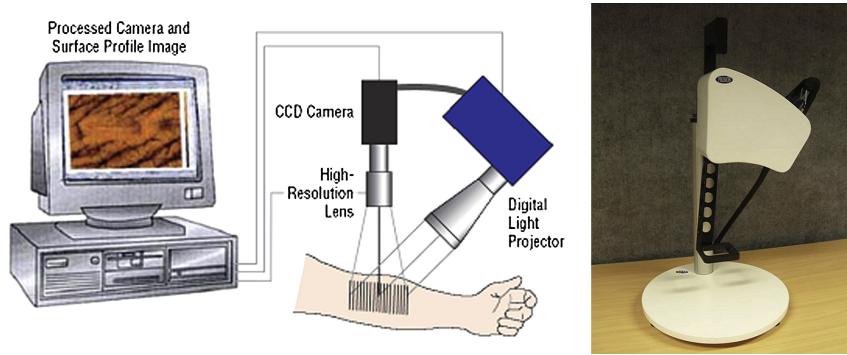


Figure 1.1: Experimental Setup of [Bloemen et al. \(2011\)](#)

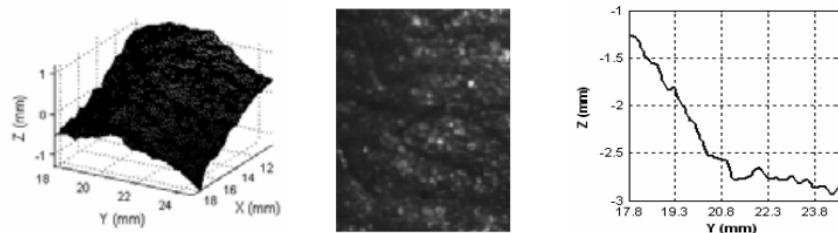


Figure 1.2: **Output of PRIMOS 3D optical scanner:** 3D surface of psoriasis lesion, 2D image of lesion, 1D profiling at first row of lesion image (from left to right); ([Fadzil et al., 2010](#))

The PASI score is assigned by means of K-means clustering. It is reported that score 1 has thickness value ranging from  $0.032\text{ mm}$  to  $0.202\text{ mm}$ . Score 2 has thickness value ranging from  $0.208\text{ mm}$  to  $0.410\text{ mm}$ . Score 3 has thickness value ranging from  $0.463\text{ mm}$  to  $0.689\text{ mm}$ . Score 4 has thickness value ranging from  $0.911\text{ mm}$  to  $2.268\text{ mm}$ . [Lee, Lee, Oh, Kim, and Kim \(2014\)](#) developed a tactile feedback system for virtual

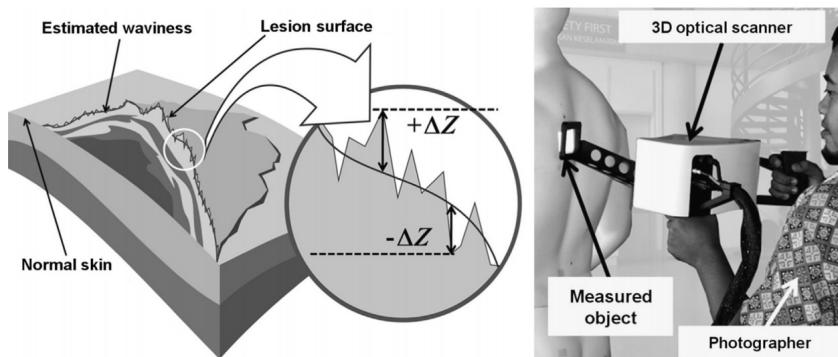


Figure 1.3: **Experimental Setup of Ahmad Fadzil et al. (2013):** cross sectional view of psoriasis lesion and estimated waviness surface and experimental setup

skin wrinkle simulation using a commercialized haptic device. To provide the tactile feedback, initially, a pair of stereo skin images were obtained by a stereo camera (field of view is  $80\text{ mm} \times 60.4\text{ mm}$ ) and disparity map of the skin wrinkles is generated using graph cuts method. The generated disparity map is sent to tactile rendering algorithm to compute a reaction force according to the user's interaction with the skin image. This force is eventually delivered to the end-user through the haptic device.

### 1.2.2 Visual SLAM

Over the last few years, SLAM using cameras has received wide attention due to simple configuration and high technical difficulties. Since this technique includes visual input, it is popularly known as *visual*

*SLAM*. Visual SLAM is an overlapping area of two different disciplines: robotics and computer vision. The term SLAM comes from the robotic society. However, long before that, the computer vision society already addressed the problem of what is called *SfM*, Structure from Motion: a technique to estimate camera motion and 3D structure of the environment in a batch manner. These set of algorithms are especially suitable for camera pose estimation in augmented reality systems because the configuration of the systems can be simple such as camera-mounted tablets or smartphones (van der Heijden, 2020; Taketomi, Uchiyama, & Ikeda, 2017).

To the best of my knowledge, the problem of 3D reconstruction of the skin or diseased skin size using handheld device and visual SLAM techniques has not received much attention yet. EKF SLAM is one of the early implementations of the SLAM algorithm. Grasa, Civera, Guemes, Munoz, and Montiel (2009) proposed an EKF based monocular SLAM to using input image sequences corresponding to a real handheld endoscope observing the abdominal cavity. The main assumptions in the proposed algorithms are scene rigidity, smooth endoscope motion and the low motion clutter.

Mahmoud et al. (2017) use PTAM like algorithm, a multi-threaded approach, to automatically and sequentially compute a 3D photogrammetric reconstruction. FAST features with ORB descriptor is used to describe the landmarks. Further to reduce the outliers, a criterion of having Shi-Tomasi score greater than 100 is set. Further bundle adjustment is used to map 3D positions accurately. Using this, they were successfully a see-through AR solution on the patient using visual SLAM.

Further, Solin, Cortes, Rahtu, and Kannala (2018); Fang, Zheng, and Deng (2016); Mourikis and Roumeliotis (2007) rely on integration of inertial odometry by means of IMU to provide more robust results. Cheng, Sun, and Meng (2019) and Alcantarilla, Yebes, Almazán, and Bergasa (2012a); Bang, Lee, Kim, and Lee (2017) discuss the application of *optical flow*, which describes the 2D motion of the pixels in the image and the *scene flow*, which describes the 3D motion of the points in the scene, to improve the robustness of the visual SLAM problem.

### 1.3 Organization of the Thesis

This thesis is organized into two parts *i)* *Design of the Stereo System* and *ii)* *3D Reconstruction using Visual SLAM*. While Chapters 2 to 4 deal with Design of the Stereo System and Chapters 5 to 7 deal with the second part.

**Chapter 2** provides the reader with background information on lenses, camera models, two camera geometry. Further, this chapter also walks the reader through the mathematical formulation of the single and two view geometry and calculation of the geometrical and optical requirements.

**Chapter 3** presents a detailed analysis of the various parameters affecting the performance of the stereo camera setup. Further, this chapter also discusses the implementation of a simulator which calculates the relevant camera geometry and optical parameters.

**Chapter 4** describes the experiments conducted with the stereo camera system. Further, this chapter elucidates the motivation behind the experiments, experimental protocol and methods and results obtained. Finally, these obtained results are discussed in the discussion section, and recommendations are provided for future work.

**Chapter 5** provides background information and the formulation of the *error-state extended Kalman filtering* and *SLAM problem*. The theory developed in this chapter shall be used for the implementation.

**Chapter 6** presents the *system model* and the *measurement model* of the stereo-camera system used by the ES-EKF SLAM algorithm. Further, this chapter also discusses the implementation of ES-EKF SLAM algorithm by highlighting the data simulation and collection strategy.

**Chapter 7** describes the experiments conducted with the SLAM algorithm. Further, this chapter eluci-

dates the motivation behind the experiments, experimental protocol and methods and results obtained. Finally, these obtained results are discussed in the discussion section, and recommendations are provided for future studies.

Finally, **Chapter 8** draws conclusions of the overall system where the main research goal and sub-goals formulated in this chapter are addressed and answered.



## Part I

### *Design of the Stereo System*



# Chapter 2

## Background

The purpose of this chapter is to provide background information on lenses, camera models, two camera geometry. Further, this chapter walks the reader through the mathematical formulation of the single and two view geometry. Refer Appendix A for the definitions of the terms used.

### 2.1 Field of View

The field of view (FoV) is the area under inspection that the camera can acquire at a given working distance. For a given camera, FoV depends on the working distance and angle of view (AoV), which in turn depends upon sensor dimensions and focal length. AoV can be calculated along three directions horizontal, vertical, and diagonal using respective sensor dimensions, focal length, and pixel size. In the fig. 2.1, the apex angle of the  $\Delta AOB$  (apricot) is the horizontal angle of view and the same of the  $\Delta AOD$  (aquamarine) is the vertical angle of view. The grey coloured rectangle is the corresponding field of view at the working distance  $\zeta$ . The fig. A.1b shows a more intuitive drawing of the field of view. Mathematically AoV and FoV defined as,

$$\text{AoV} = 2 \arctan\left(\frac{\text{sensor dimension}}{2f}\right) \quad (2.1)$$

$$\text{FoV} = 2\zeta \tan\left(\frac{\text{AoV}}{2}\right) \quad (2.2)$$

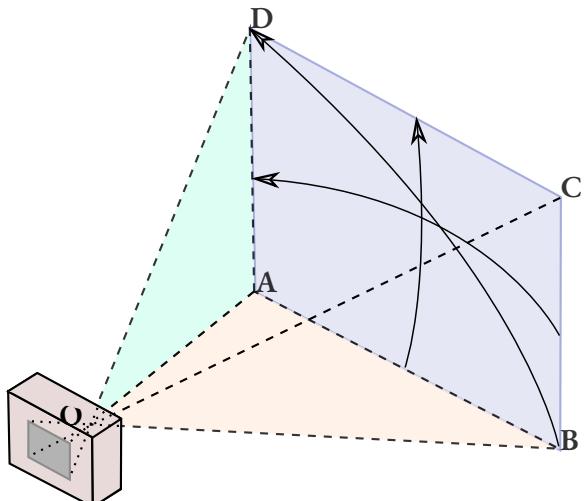


Figure 2.1: **Field of View:** The angle of view and the field of view of a camera, showing horizontal, vertical, and diagonal angle measured at a distance  $\zeta$ ; adapted from [Wikipedia contributors \(2020\)](#).

### 2.2 Depth of Field

The distance between nearest and farthest objects that are in acceptably sharp focus is known as the depth of field (DoF). The depth of field can be calculated based on focal length, working distance, the acceptable circle of confusion (coc) size, and aperture. Mathematically, it is the distance between far point and near point.

## 2.3 Coordinate Systems and Transformations

Coordinate systems are used to uniquely determine the position of the geometric elements on a manifold. There exists infinitely many such coordinate systems to uniquely identify such elements. These coordinate systems are related by a relation called *coordinate transformation*.

### 2.3.1 Coordinate Systems in Computer Vision

In computer vision for measuring the position and orientation of the objects mainly, two coordinate systems are involved *viz.*,

**Camera Coordinate System:** A coordinate system along the optical axis and the coordinates of the image plane. From now on CCS is denoted as  $\vec{x} = [x \ y \ z]^\top$ , unless specified otherwise. Note that,  $z$ -axis is aligned with the optical axis, i.e., orthogonal to the image plane. It is abbreviated as ‘ccs’ throughout the document.

**World Coordinate System:** A reference coordinate system denoted as  $\vec{X} = [X \ Y \ Z]^\top$ . It is abbreviated as ‘wcs’ throughout the document.

### 2.3.2 Coordinate Transforms

This section quickly reflects upon how the transformations between two coordinate systems can be achieved. The coordinate transforms are of central importance in robotics and vision community. In this section, two widely used methods, *i) rotation and translation matrices, ii) quaternions* are discussed.

#### Rotation and Translation Matrices

Coordinate systems are related by a set of linear transformations: translation  $\vec{t}$  and a rotation  $\mathbf{R}$ . Hence, mathematically,

$$\vec{x} = \mathbf{R}(\vec{X} - \vec{t}) \quad (2.3)$$

The rotation matrices can be denoted by  ${}^A\mathbf{R}_B$ , which represents the orientation of frame B relative to frame A. The translation matrices can be denoted by  ${}^A\vec{t}_B$ , which represents the translation of frame B w.r.t. to frame A.

Consecutive rotations can be modelled by matrix-matrix multiplication of rotation matrices, hence,

$${}^C\mathbf{R}_B = {}^C\mathbf{R}_A \otimes {}^A\mathbf{R}_B = {}^C\mathbf{R}_A {}^A\mathbf{R}_B \quad (2.4)$$

This multiplication is *not* commutative.

#### Quaternions

If two complex numbers  $A = a + bi$  and  $C = c + di$ , then constructing  $Q = A + Cj$  and defining  $k = ij$  yields a number in the space of quaternions  $\mathbb{H}$ , as defined by Hamilton in 1843,

$$Q = a + bi + cj + dk \in \mathbb{H}$$

where  $\{a, b, c, d\} \in \mathbb{R}$ , and  $\{i, j, k\} \in \mathbb{C}$  are three imaginary unit numbers defined so that

$$i^2 = j^2 = k^2 = ijk = -1$$

from which it can be derived

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j$$

In practice, the the quaternion  $Q = q_w + q_x i + q_y j + q_z k$  is generally denoted as,

$$\mathbf{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix}$$

It is noticeable that, while regular complex numbers of unit length  $z = e^{i\theta}$  can encode rotations in the 2D plane, *extended complex numbers* or quaternions of unit length  $\mathbf{q} = e^{(u_x i + u_y j + u_z k) \frac{\theta}{2}}$  encode rotations in the 3D space (Solà, 2017).

Two conventions for quaternions exist, Hamilton and JPL. The key difference between these two conventions lies in the relation between the three imaginary bases. In Hamilton convention,  $ijk = -1$ , while JPL defines  $ijk = 1$ . As consequences, the multiplication of quaternions and the transformation between quaternions and other rotation parametrizations differ with different quaternion conventions. It is explicitly mentioned that this document consistently uses Hamiltonian notation.

To represent the 3D orientation, a quaternion encodes the axis-angle representation. To parametrize 3D rotation using 4 parameters we need to have a constraint which follows from the requirement that the rotation axis has unit length. Thus, the quaternions must also have unit length, such that,

$$\|\mathbf{q}\| = q_w^2 + q_x^2 + q_y^2 + q_z^2 = 1 \quad (2.5)$$

Similar to rotation matrices, consecutive rotations can be modelled by quaternion product. It is important to note that quaternion product is *not* commutative. The quaternion product of quaternions  $\mathbf{p} = [p_w \ p_v]^\top$  and  $\mathbf{q} = [q_w \ q_v]^\top$  is defined as,

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - p_x q_x - p_y q_y - p_z q_z \\ p_w q_x + p_x q_w + p_y q_z - p_z q_y \\ p_w q_y - p_x q_z + p_y q_w + p_z q_x \\ p_w q_z + p_x q_y - p_y q_x + p_z q_w \end{bmatrix} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^\top \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + \mathbf{p}_v \times \mathbf{q}_v \end{bmatrix} \quad (2.6)$$

This product can be re-written in vector-matrix multiplication scheme as,

$$\mathbf{p} \otimes \mathbf{q} = [\mathbf{p}]_L \mathbf{q} = \begin{bmatrix} p_w & -p_x & -p_y & -p_z \\ p_x & p_w & -p_z & p_y \\ p_y & p_z & p_w & -p_x \\ p_z & -p_y & p_x & p_w \end{bmatrix} \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \quad (2.7)$$

The theory developed in this section is heavily used as the basis for ES-EKF SLAM.

## 2.4 Camera Models

A camera is a mapping between the 3D world and a 2D image. Here we use the model principally designed for CCD like sensors (Hartley & Zisserman, 2004) and is used most often in applications and academic research, i.e., the pinhole model (Sturm, Ramalingam, Tardif, Gasparini, & Barreto, 2011). The following subsection briefly describes the *pinhole camera model*.

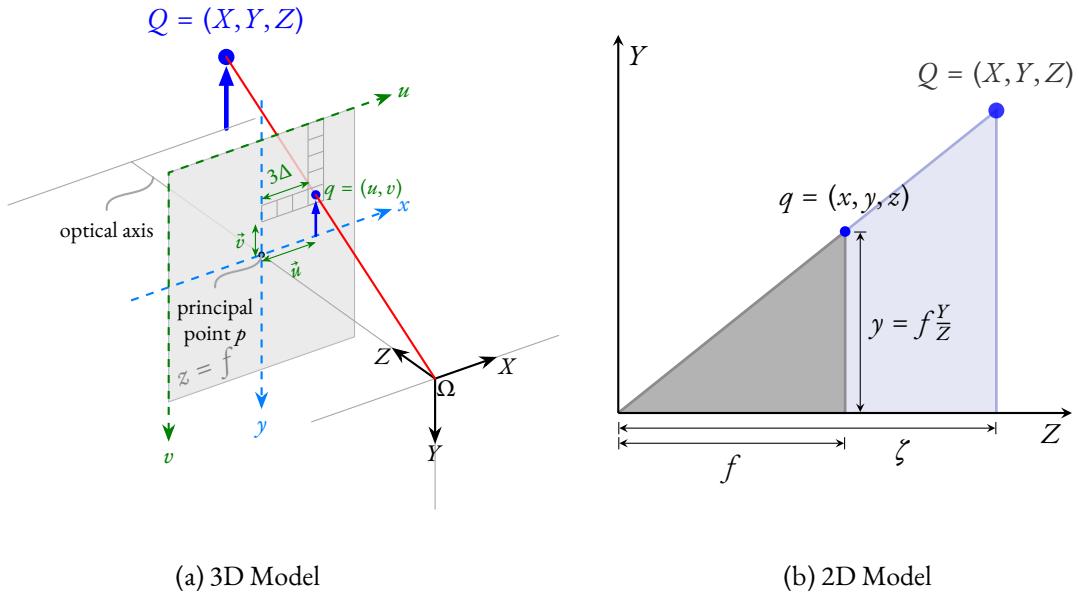


Figure 2.2: **Pin-hole camera geometry:**  $\Omega$  is a camera centre,  $p = (p_x, p_y)$  is the **principal point** or **image centre**,  $(u, v)$  are **pixel coordinates**,  $f$  is a focal length and  $\Delta$  is a pixel size. Adapted from cfr (2016); Hartley and Zisserman (2004).

#### 2.4.1 The Pinhole Camera Model

Figure 2.2 shows the pinhole camera model. According to this model, a point light source  $Q$  at position  $(X, Y, Z)$  will lie on the intersection of a ray from the point  $Q$  to the origin  $\Omega$  and the image plane.

Using basic proportionality theorem in fig. 2.2b, following equations can be written,

$$\frac{x - p_x}{X} = \frac{y - p_y}{Y} = \frac{f}{Z} \quad (2.8)$$

From eq. (2.8), it can be observed that the point  $(X, Y, Z)^\top$  is mapped to the point  $(f\frac{X}{Z} + p_x, f\frac{Y}{Z} + p_y, f)^\top$ . Hence, the central projection mapping ( $\mathbb{R}^3 \mapsto \mathbb{R}^2$ ) can be mathematically defined as (Hartley & Zisserman, 2004),

$$(X, Y, Z)^\top \mapsto \left( f\frac{X}{Z} + p_x, f\frac{Y}{Z} + p_y \right)^\top \quad (2.9)$$

In homogeneous coordinates, above equation can be conveniently written as,

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Xp_y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2.10)$$

More concisely using pixel coordinates  $\vec{p} = [u \ v]^\top$ ,

$$\vec{p} = \mathbf{K} [\mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 1}] \vec{X} \quad (2.11)$$

Where,  $\mathbf{K}$  is a camera *calibration matrix* containing the intrinsic parameters of the camera.

The mathematical formulations above are in Camera Coordinate System (CCS) but, in the real world camera typically has its own frame of reference, called World Coordinate System (WCS). Transformation between these coordinate systems involve a rotation and a translation which can be written as,

$${}^{CCS}\vec{X} = {}^{CCS}\mathbf{R}_{WCS} {}^{WCS}\vec{X} + {}^{CCS}\vec{t}_{WCS} \quad (2.12)$$

In homogeneous coordinates,

$${}^{CCS}\underline{\vec{X}} = {}^{CCS}\mathbf{T}_{WCS} {}^{WCS}\underline{\vec{X}} \quad (2.13)$$

Where,

$${}^{CCS}\mathbf{T}_{WCS} = \left[ \begin{array}{c|c} {}^{CCS}\mathbf{R}_{WCS} & {}^{CCS}\vec{t}_{WCS} \\ \hline 0 & 1 \end{array} \right] \quad (2.14)$$

Using eqn. (2.11) and eqn. (2.13),

$$\underline{\vec{p}} = \mathbf{K} [{}^{CCS}\mathbf{R}_{WCS} \quad {}^{CCS}\vec{t}_{WCS}] {}^{WCS}\underline{\vec{X}} \quad (2.15)$$

The rotation matrix and translation vector are called the *extrinsic camera parameters*.

Throughout above discussion the underlying assumption has been that the linear model is an accurate model of the imaging process. This implies that world point, image point and optical centre are collinear, and world lines are imaged as lines etc. For real lenses i.e., non-pinhole this assumption will not hold. Hence, we need to adapt the camera model to include non-linear lens distortions, viz., radial and tangential distortions. [van der Heijden \(2019b\)](#) and [Hartley and Zisserman \(2004\)](#) discuss this in more detail.

## 2.5 Epipolar Geometry

The epipolar geometry is the intrinsic projective geometry between two views which is independent of scene structure, and only depends on the cameras' internal parameters and relative pose ([Hartley & Zisserman, 2004](#); [Nister, 2004](#)). This intrinsic geometry is encapsulated by the *fundamental matrix*  $\mathbf{F}$  and shall be introduced in sections below.

### 2.5.1 The Geometry of Two Cameras

A 3D point  $X$  is imaged by two cameras in their own coordinated systems represented as  ${}^1\vec{X}$  and  ${}^2\vec{X}$  respectively. Taking inspiration from section 2.4.1, equations for  ${}^1\vec{X}$  and  ${}^2\vec{X}$  can be written as,

$${}^1\vec{X} = {}^1\mathbf{R}_2 + {}^2\vec{X} + {}^1\vec{t}_2 \quad (2.16)$$

$${}^2\vec{X} = {}^1\mathbf{R}_2^\top ({}^1\vec{X} - {}^1\vec{t}_2) = {}^2\mathbf{R}_1 + {}^1\vec{X} + {}^2\vec{t}_1 \quad (2.17)$$

The origin of camera 1 and 2 are separated by baseline distance  $b$ , also known as, interocular distance. This implies  ${}^1\vec{t}_2$  is equal to the baseline distance.

### 2.5.2 The Epipolar Constraint

The fig. 2.3 shows the geometry of the two camera setup. The point  ${}^1e$  in image  $I_1$  is a function of the position of the second camera. Consider Camera 1. The image point  ${}^1p$  is a function of world point  $P$ . The camera centre,  ${}^1e$  and  ${}^1p$  define *epipolar plane* and hence the intersection with second image plane defines the second *epipolar line*. Hence, the conjugate point  ${}^2p$  of any point  ${}^1p$  lying on the first epipolar line should lie on this line. This is a very fundamental and important geometric relationship: *given a*

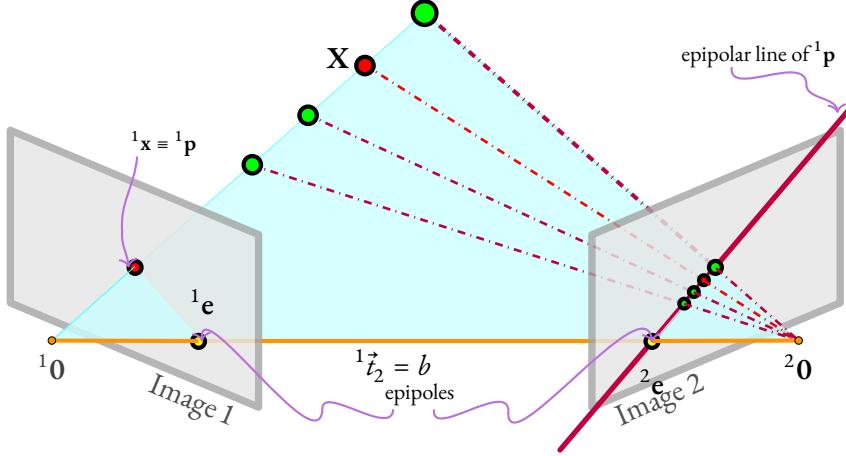


Figure 2.3: **The Geometry of Two Cameras:** Epipolar geometry showing the two cameras with associated coordinate frames  ${}^1\mathbf{0}$  and  ${}^2\mathbf{0}$  and image planes. The world point  $\mathbf{X}$  and the two camera centres form the **epipolar plane**, and the intersection of this plane with the image planes form **epipolar lines**; adapted from van der Heijden (2019d).

*point in the first image its conjugate is constrained to lie along a line in the other image* (Corke, 2017). This **epipolar constraint** is guaranteed by the *Fundamental Matrix F* in homogeneous pixel coordinates and by *Essential Matrix E* in normalized image coordinates (Corke, 2017). Mathematically these relationships can be concisely expressed as

$${}^2\vec{X}^\top \mathbf{E} {}^1\vec{X} = 0 \quad (2.18)$$

$${}^2\vec{p}^\top \mathbf{F} {}^1\vec{p} = 0 \quad (2.19)$$

Due to orthogonality of vectors  $({}^1\vec{t}_2 \times {}^1\vec{X})$  and  $({}^1\vec{X} - {}^1\vec{t}_2)$ , Essential matrix can be expressed as,

$$\mathbf{E} := {}^2\mathbf{R}_1 \mathbf{T} \quad (2.20)$$

Here  $\mathbf{T}$  is a cross product operator of  ${}^1\vec{t}_2 = [{}^1t_x \quad {}^1t_y \quad {}^1t_z]$ , defined as,

$$[{}^1\vec{t}_2]_\times = \mathbf{T} = \begin{bmatrix} 0 & -{}^1t_z & {}^1t_y \\ {}^1t_z & 0 & -{}^1t_x \\ -{}^1t_y & {}^1t_x & 0 \end{bmatrix} \quad (2.21)$$

Using, eqn. (2.15) and (2.18) it can be easily shown that,

$$\mathbf{F} := \mathbf{K}_2^{-\top} \mathbf{E} \mathbf{K}_1^{-1} \quad (2.22)$$

## 2.6 Depth Perception

According to [The National Institute for Rehabilitation Engineering](#), depth perception has a very specific meaning. It means the distance straight ahead of the viewer's eye, toward or into an object or surface. By definition, depth is looking straight into a hole or tube and estimating forward distances. Perceiving depth accurately requires binocular stereoscopic vision or stereopsis.

### 2.6.1 Human Eye Depth Perception

Humans perceive depth from a variety of depth cues. These are typically classified into *binocular cues* that are based on the receipt of sensory information in three dimensions from both eyes and *monocular cues*

that can be represented in just two dimensions and observed with just one eye (Howard, 2012), as cited in [Vivid Vision](#) (n.d.). Binocular cues include stereopsis, eye convergence, disparity, and yielding depth from binocular vision through the exploitation of parallax. Monocular cues include size: distant objects subtend smaller visual angles than near objects, grain, size, and motion parallax ([Vivid Vision](#), n.d.). In this work, we rely on stereopsis and convergence as the stereoscopic viewing system designed does not depend on the context of the visual scene.

### 2.6.2 Stereoscopic Depth Perception

A passive camera system can be compared to the binocular vision of human eyes. The depth perception of an object can be determined by combining two separate 2D images from two different points of view, calculated via a method called *triangulation* ([van der Valk, 2019](#); [van der Heijden, 2019d](#)). This method uses the difference between the distance (known as *disparity*) of same interest 3D point (also known as key point or landmark) in two images. A triangle is formed (refer fig. 2.4) between the two focal points of the two cameras and the landmark. From these parameters, the distance to the scene point can be calculated. Hence, the equation for perceived depth can be written as,

$$Z = f \frac{b}{d_1 - d_2} = f \frac{b}{d} \quad (2.23)$$

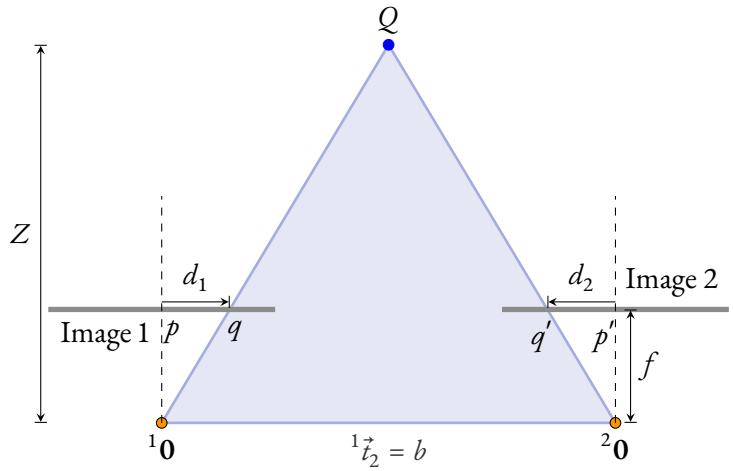


Figure 2.4: **Triangulation in a Stereo Vision system:**  $Q$  is the 3D world point imaged as  $q$  and  $q'$  in respective image planes;  $d = d_1 - d_2$  is the disparity; adapted from [Munro and Gerdelen \(2009\)](#).

### 2.6.3 Convergence

In order to look at a point, both our eyes have to converge to that point. This rotation of the eyes in order to concentrate on a single point (fixation point) in the scene space is called *convergence*. Eyes can focus on a single point at a given time. In order to focus at another point in the scene, the eyes may have to converge or diverge depending upon the location of the new fixation point with respect to the old point in the scene. The visual system may use this information to estimate the distance of the point from the observer or, in other words, the depth of that point in the scene ([Prakash, 2007](#)). Further, an optimal convergence angle can be found using the distance of the known fixation point. Hence, mathematically convergence angle can be calculated as,

$$\alpha = 2 \arctan\left(\frac{b}{2Z}\right) \quad (2.24)$$

## 2.7 Calculation of Geometrical and Optical Requirements

This section formulates the basis for calculation of optical and geometrical parameters used to design the stereo camera setup with the desired depth resolution.

### 2.7.1 Depth Resolution

The *depth resolution* of stereocamera can also be defined as the amount of change in perceived depth caused by unit change in disparity. Mathematically, this can be represented by differentiating eqn. (2.23) w.r.t. disparity and substituting  $d = \frac{Z}{bf}$ .

$$\begin{aligned}\frac{dZ}{dd} &= -\frac{bf}{d^2} \\ \left| \frac{dZ}{dd} \right| &= \frac{Z^2}{bf}\end{aligned}\quad (2.25)$$

### 2.7.2 Minimum Baseline

Minimum Baseline  $b_{\min}$ , which satisfies desired depth resolution for a specific set of focal length and working distance  $\zeta$  can be calculated by substituting  $Z$  as working distance. Using basic proportionality theorem to fig. 2.2b,

$$\zeta = f \times \frac{\text{size of the object}}{\text{size of the image}} \quad (2.26)$$

Using above equation in the eqn. (2.25),

$$b_{\min} = f \times \frac{dd}{dZ} \times \left( \frac{\text{size of the object}}{\text{size of the image}} \right)^2 \quad (2.27)$$

## 2.8 3D Landmark Estimation from Multiple Images

This section discusses the Minimum Mean Square Error (MMSE) estimation technique for estimating the 3D position of the landmark as developed by [van der Heijden \(2019a\)](#). It is assumed that both the cameras are fully calibrated, i.e., intrinsic parameters and pose of the cameras w.r.t. reference coordinate system (0) is known. The objective, here, is to estimate the 3D coordinates of the landmark in the reference coordinate system.

MMSE estimation technique is based on the observation model of the system. Here, a linear estimator with additive Gaussian noise is considered. The observation model is hypothesized as,

$$\vec{z} = \mathbf{H}^0 \vec{X} + \vec{n} \quad (2.28)$$

where,

${}^0\vec{X}$  = unknown parameter vector, i.e., 3D position of the landmark represented in frame 0

$\mathbf{H}$  = measurement matrix describing linear behaviour of the sensor system

$\vec{n}$  = zero mean Gaussian distributed sensor noise with covariance matrix  ${}^2\mathbf{C}_n$

$\vec{z}$  = observed measurement vector

The prior knowledge of the parameter of interest, here, 3D position is quantified by means of the expectation  ${}^0\hat{\vec{X}}(t | t-1)$  with the uncertainty of  $\mathbf{C}(t | t-1)$ . In case of no prior knowledge, an conservative estimate is assumed such that,

$$\mathbf{C}(t | t-1) = \sigma_{t|t-1}^2 \quad \text{with} \quad \sigma_{t|t-1} \gg \text{depth range} \quad (2.29)$$

The following set of equations define the MMSE estimation algorithm:

---

**Algorithm 2.1:** MMSE Estimation Process

---

- 1  $\mathbf{S} = \mathbf{H}\mathbf{C}(t | t-1)\mathbf{H}^\top + \mathbf{C}_n$  // innovation matrix
  - 2  $\mathbf{G} = \mathbf{C}(t | t-1)\mathbf{H}^\top \mathbf{S}^{-1}$  // Kalman gain matrix
  - 3  ${}^0\hat{\vec{X}}(t | t) = {}^0\hat{\vec{X}}(t | t-1) + \mathbf{G}(\vec{z} - \mathbf{H}{}^0\vec{X}(t | t-1))$  // posterior estimate
  - 4  $\mathbf{C}(t | t) = \mathbf{C}(t | t-1) - \mathbf{G}\mathbf{S}\mathbf{G}^\top$  // posterior covariance matrix
- 

Above set of equations are only valid if the observation model is linear and is in accordance with eqn. (2.28). After rearranging and solving the observation model (eqn. (2.15)) can be rewritten as:

$$\vec{z}_1 := \begin{bmatrix} {}^1\hat{p}_1 & \vec{k}_3^\top & \vec{k}_1^\top \\ {}^1\hat{p}_2 & \vec{k}_3^\top & \vec{k}_2^\top \end{bmatrix} {}^1\vec{t}_0; \quad \mathbf{H}_1 := \begin{bmatrix} \vec{k}_1^\top & {}^1\hat{p}_1 & \vec{k}_3^\top \\ \vec{k}_2^\top & {}^1\hat{p}_2 & \vec{k}_3^\top \end{bmatrix} {}^1\mathbf{R}_0; \quad \vec{n}_1 := \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} {}^1Z \quad (2.30)$$

Where  ${}^1\hat{p}_1$  is the first element of the  ${}^1\vec{p}_1$  and  $\varepsilon_1$  and  $\varepsilon_2$  are localization errors in the estimation of  ${}^1\hat{p}_1$  and  ${}^1\hat{p}_2$  respectively.  $k_1, k_2, k_3$  are the respective rows of  $\mathbf{K}_1$  and  ${}^1Z$  is the z-component of the  ${}^1\vec{X}$ .

As the error propagation in eqn. (2.30) is linear with  ${}^1Z$ , covariance matrix of  $\vec{n}_1$  becomes,

$$\mathbf{C}_{n_1} = {}^1Z^2 \sigma^2 \mathbf{I}$$

Similar theory can be used to build a model for knowledge coming from the second camera. This knowledge can be combined using a set of equations defined in algorithm 2.1 in a well balanced manner.



# Chapter 3

## Design and Implementation

This chapter presents a detailed analysis of the various parameters affecting the performance of the stereo camera setup. Initially, the adjustable parameters of the system are identified. These parameters are further divided into two classes viz., parameters affecting the accuracy and dependent parameters. Further, this chapter also discusses the implementation of a simulator which calculates the relevant camera geometry and optical parameters.

Table 3.1: **Effect of Increasing Adjustable Parameters on the Performance Parameters:** ↑ represents a increase, ↓ represents a decrease, and × represents no change

Parameter (↑)	Disparity Values	Perceived Depth	FOV	Overlap Area	Hyper Focal Distance	Depth Resolution <sup>1</sup>
Sensor Resolution	↑	↑	↑	↑	×	×
Pixel Size	↓	↓	↑	↑	×	↓
Focal Length	↑	↑	↓	↓	↑	↑
f-number	×	×	×	×	↓	×
Circle of Confusion	×	×	×	×	↓	×
Baseline	↑	↑	↓	↓	×	↑
Working Distance	↓	↓	↑	↑	×	↑
Convergence Angle	↓	Shift	×	×	×	×

### 3.1 Identification of the Geometric and Optical Parameters

A thorough study was conducted to identify the adjustable geometric and optical parameters and their effects on the performance parameters of the stereo camera system. This section initially reports these adjustable parameters and performance parameters and later tabulates their effect on the performance parameters. Definitions of these parameters are present in Appendix A.

<sup>1</sup>small number is good, hence for small number ↑ is used.

## List of Optical Parameters

## List of Geometric Parameters

- |   |   |
|---|---|
| <ol style="list-style-type: none"> <li>1. Sensor Size or Sensor Diameter</li> <li>2. Sensor Resolution</li> <li>3. Pixel Size</li> <li>4. Pixel Pitch</li> <li>5. Focal Length</li> <li>6. <i>f</i>-number or aperture</li> <li>7. Circle of confusion</li> </ol> | <ol style="list-style-type: none"> <li>1. Baseline</li> <li>2. Working distance</li> <li>3. Stereo angle</li> </ol> |
|---|---|

It is important to note that due to standard availability of camera sensors, optical parameters are adjustable but fixed for a particular sensor, hence there is a limit on their adjustability. Hence, there is quite a freedom to vary geometrical parameters. Table 3.1 depicts the effect of adjusting various geometric and optical parameters on the performance parameters.

## 3.2 Calculation of the Stereo-geometry

The research group Robotics and Mechatronics had two FriendlyElec CAM1320<sup>2</sup> cameras as per design space exploration done by Grimm (2020).

This camera module uses OmniVision's CMOS OV13850 module<sup>3</sup>. The specifications of interest are listed below.

- focal length: 2.94 mm
- images size: 4224 pixels × 3136 pixels or 4.8 mm × 3.7 mm
- pixel size: 1.12 μm × 1.12 μm
- optical format: 1/3 inch

The requirement of the camera system is to have a depth resolution of at least 0.1 mm with large combined view field. For fixed optical parameters, this overlap can be varied by means of varying baseline and tilting both cameras towards each other.

Following calculations are made considering the size of the object to be reconstructed using this system is around 400 mm. Using eqn. (2.27) and one of the image dimension,

$$b_{\min} = 2.94 \times \frac{1.12 \times 10^{-3}}{0.1} \times \left( \frac{400}{4.8} \right)^2 = 228.667 \approx 230 \text{ mm}$$

The optimum working distance for this focal length and object size using eqn. (2.26) evaluates to be 245 mm.

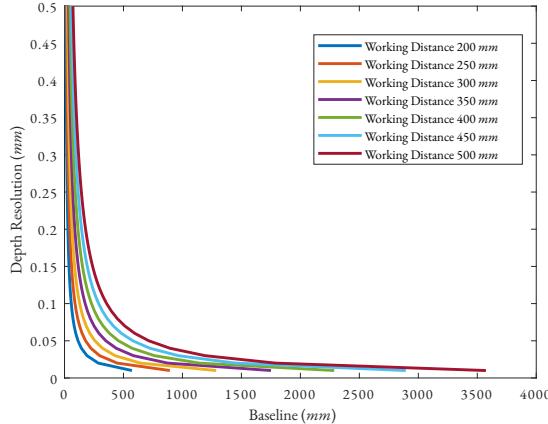
## 3.3 The Camera Geometry Visualiser App

A camera geometry visualiser app is developed using MATLAB app designer to visualise stereo geometry and various parameters graphically. Figures 3.2a and 3.2c shows the screen capture of the various visual-

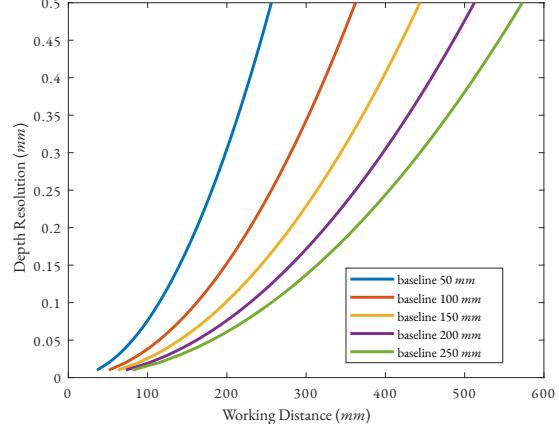
---

<sup>2</sup>[https://www.friendlyarm.com/index.php?route=product/product&product\\_id=228](https://www.friendlyarm.com/index.php?route=product/product&product_id=228)

<sup>3</sup>[http://www.t-firefly.com/download/firefly-rk3288/peripherals/Sensor\\_OV13850-G04A\\_OmniVision\\_Specification\(V1.1\).pdf](http://www.t-firefly.com/download/firefly-rk3288/peripherals/Sensor_OV13850-G04A_OmniVision_Specification(V1.1).pdf)



(a) Baseline vs. Depth Resolution;



(b) Working Distance vs. Depth Resolution

Figure 3.1: **Depth Resolution Plots:** Plots of depth resolution as a function of (a) baseline and; (b) working distance

izations from the tabular interface of the app. This app has a tabular interface, which is divided into three tabs, viz., *Camera Parameters Plot*, *Projected Points* and *Reconstructed Points*.

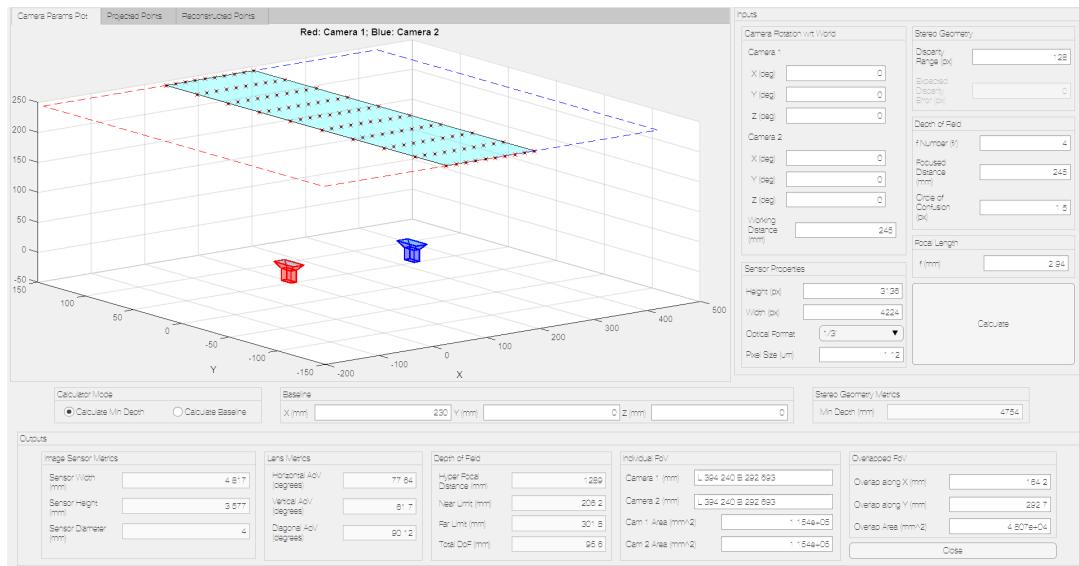
This app calculates relevant parameters to obtain the desired depth resolution. These parameters are as listed below. Based on these calculations a stereo geometry is sketched in the Camera Parameters Plot tab. (refer fig. 3.2a):

1. Calculation of optimum distance from the camera
2. Calculation of minimum baseline required to achieve the desired depth resolution
3. Calculation of optical and geometrical parameters. These parameters are categorised in the following classes
  - (a) Image Sensor Metrics: sensor width, sensor height, sensor diameter
  - (b) Lens Metrics: horizontal angle of view, vertical angle of view, diagonal angle of view
  - (c) Depth of Field: hyperfocal distance, near and far distance, total depth of field
  - (d) Individual Field of View of both cameras
  - (e) Overlapped Field of View of both cameras

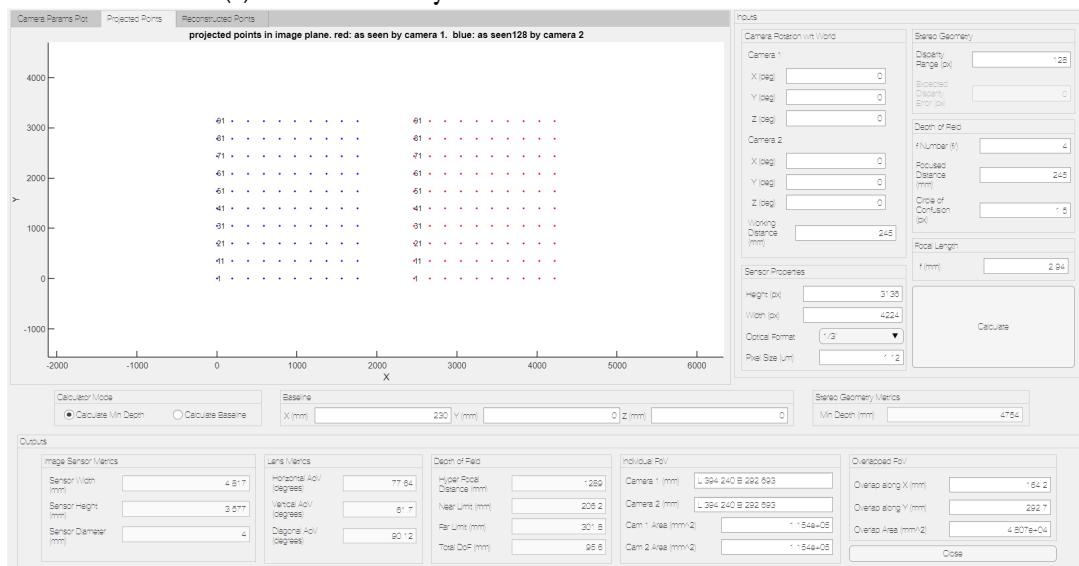
The second tab, Projected Points, shows how each camera sees a given array of points. This is shown in fig. 3.2b.

Last tab, Reconstructed Points, plots the reconstruction of 3D points from the 2D projection of above point. For this calculation algorithm 2.1 is used. The uncertainty of reconstruction of each point is also visualized by means of the ellipse.

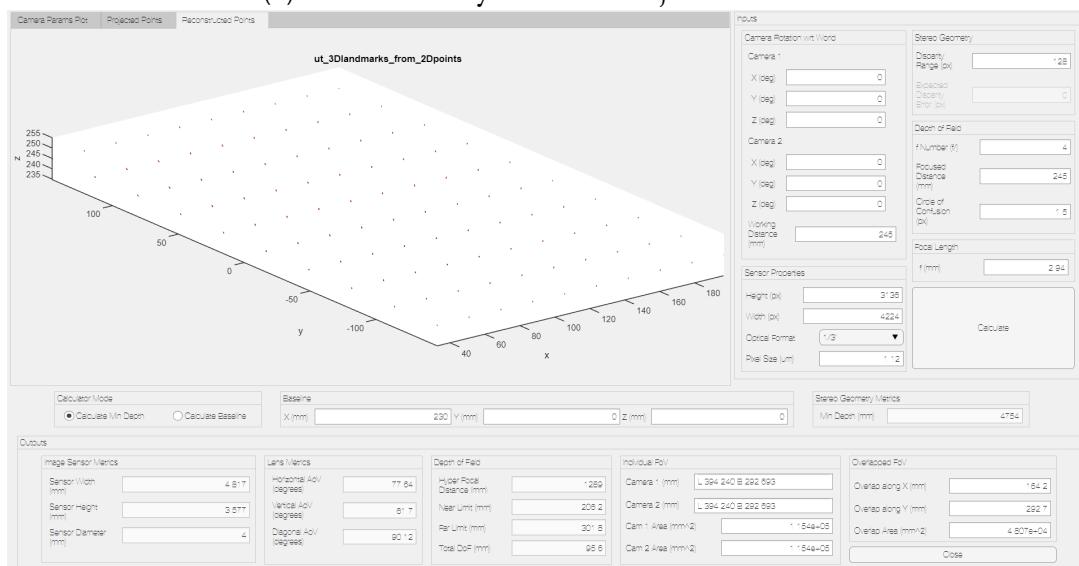
Based on the above simulator and calculations made in section 3.2, Grimm (2020) has developed a stereo camera setup as shown in fig. 3.3.



(a) Stereo Geometry Visualiser: Camera Parameters Plot tab



(b) Stereo Geometry Visualiser: Projected Points tab



(c) Stereo Geometry Visualiser: Reconstructed Points tab

Figure 3.2: Camera Simulator App

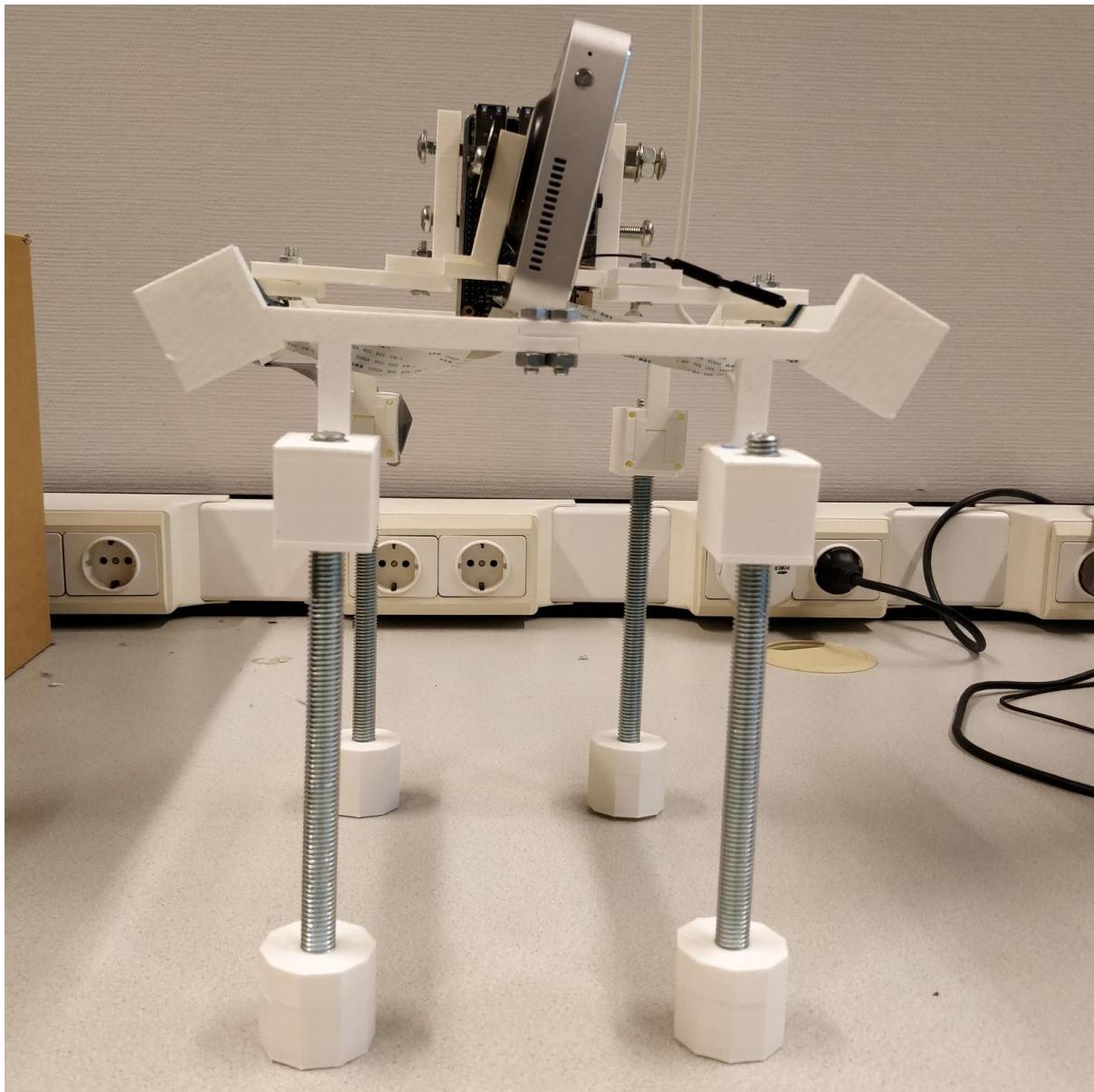


Figure 3.3: Stereo Camera Setup



# Chapter 4

## Experiments

This chapter describes the experiments conducted with the system designed in the previous chapters. Further, this chapter elucidates the motivation behind the experiments, experimental protocol and methods and results obtained. Finally, these obtained results are discussed in the discussion section.

For preliminary experiments performed using this camera system, a report by [Grimm \(2020\)](#) should be referred.

### 4.1 Motivation and Goals

The motivation behind conducting the experiments is to evaluate the ability of the stereo camera to differentiate between the distances in the order of  $0.25\text{ mm}$ . Distances in this range are expected while assigning the score using PASI ([Prof. Dr. med. Kristian Reich, 2013](#)). The secondary motivation is to perform the experiments in a systematic and reproducible manner.

### 4.2 Methods

For performing the experiments in a reproducible manner, a 3D printed workpiece is used. Appendix C shows an isometric view and a technical drawing of this workpiece, whereas, fig. 4.1 shows an image of the workpiece taken using a cellphone camera. Due to the shiny and reflective surface of the workpiece, it was decided not to project patterns on the workpiece, but rather, draw patterns on it using colour markers. Two iterations of the same experiment are performed, with different patterns drawn on the workpiece. These patterns are reported in the fig. 4.2, referred from now as a yellow pattern and white pattern, respectively.

As previously shown, the field of view of our cameras is well below  $95^\circ$ , the *Stereo Camera Calibrator App* from MATLAB's Computer Vision Toolbox is used to calibrate the stereo camera system. A planar chequerboard pattern (each square  $10\text{ mm} \times 10\text{ mm}$ , fig. 6.1a) is used for calibration.

To achieve the stereo matching, DAISY descriptors are calculated as per ([Tola, Lepetit, & Fua, 2008, 2010](#)). The C++ code for the same is made available under BSD License by authors at <https://www.epfl.ch/labs/cvlab/software/descriptors-and-keypoints/daisy>. This design choice was made by [Grimm \(2020\)](#) due to wide baseline of the camera.

A MATLAB app developed by [Grimm \(2020\)](#) to facilitate a bridge between the stereo camera system and MATLAB Stereo Camera Calibrator app. It is important to note that this app uses a deprecated MATLAB class `tcpip` which was a part of Instrument Control Toolbox. This legacy class has been removed from MATLAB R2020a and makes app no longer usable with the newer versions. Hence, it is advised to use



Figure 4.1: Workpiece used for Reconstruction: image taken using cellphone camera



Figure 4.2: Patterns Drawn on the Workpiece: taken using cellphone camera

MATLAB R2019a. Further, the DAISY library is written for 32-bit (x86) architecture and needs to be migrated to 64-bit (x64) architecture.

Following versions of tools are used:

- **IDE:** Visual Studio 2019
- **C++ Compiler:** MSVC C++
- **C++ Standard:** C++14
- **OpenCV:** 3.4.3<sup>1</sup>
- **DAISY:** 1.8.1<sup>2</sup>

<sup>1</sup>This is pre-built 64-bit version availed from OpenCV website. This version is not directly available via vcpkg.

<sup>2</sup>migrated to 64-bit manually

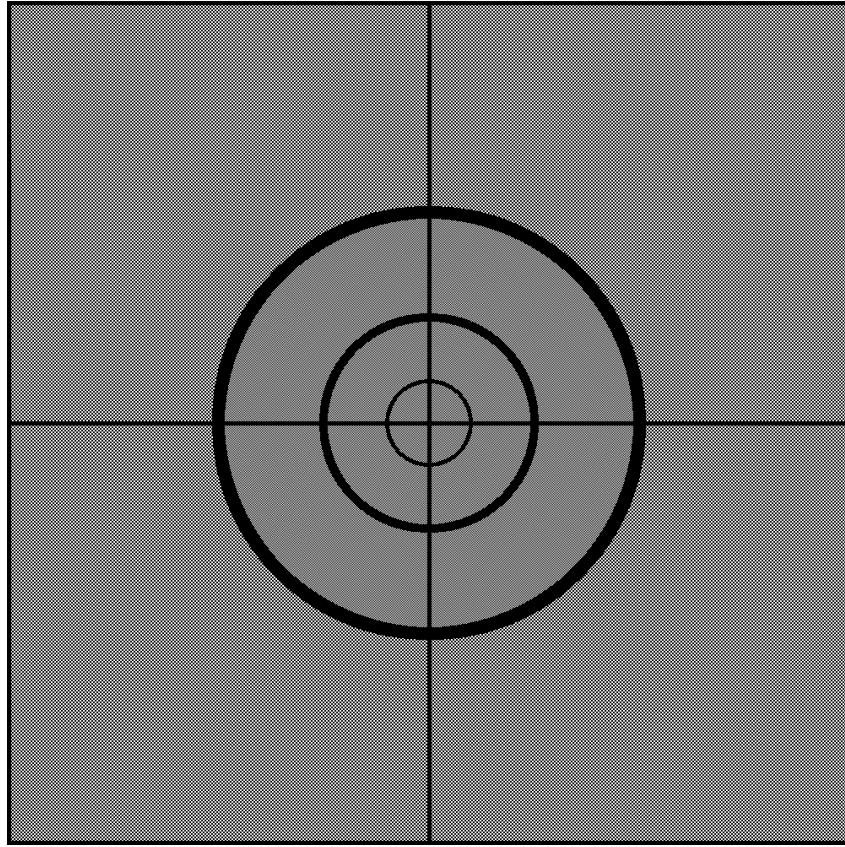


Figure 4.3: High Contrast Pattern for Auto-Focus

Further, as the camera module uses *contrast-detection*, a passive auto-focus technology, for autofocus. To achieve good focus a object with good contrast and good lighting conditions are required. Further, the application developed by Grimm (2020) uses central  $100 \times 100$  pixels for auto-focus. Hence, it is recommended to have a good contrast in this specific area. For auto-focus purposes, a printout of fig. 4.3 is used.

To evaluate the ability of the stereo camera to differentiate between the distances plane-to-plane distance for embossed surfaces and ground plane is calculated. To do this CloudCompare is be used.

### 4.3 Results

Upon calibrating the camera following results are obtained:

Table 4.1: Camera Calibration Results

Actual	Estimated	
	Yellow Pattern	White Pattern
Baseline	210 mm	211.5 mm
Angle between cameras	54.4 °	55.29 °
Mean reprojection Error	—	0.52 pixels
		0.72 pixels

The fig. 4.4 shows the montage of an undistorted image taken from the left and the right camera, respectively. The figs. 4.5 and 4.6 show the meshes built from the reconstructed 3D dense point cloud. The

Table 4.2: **Reconstructed Heights**

Actual (mm)	Calculated (mm)	Std. Dev. (mm)
0.2	0.24	0.020000
0.3	0.29	0.000019
0.4	0.38	0.000007
0.5	0.45	0.003449
0.6	0.56	0.002000
0.7	0.66	0.100000

colour of the left image is transferred to the point cloud for more realistic visualization.

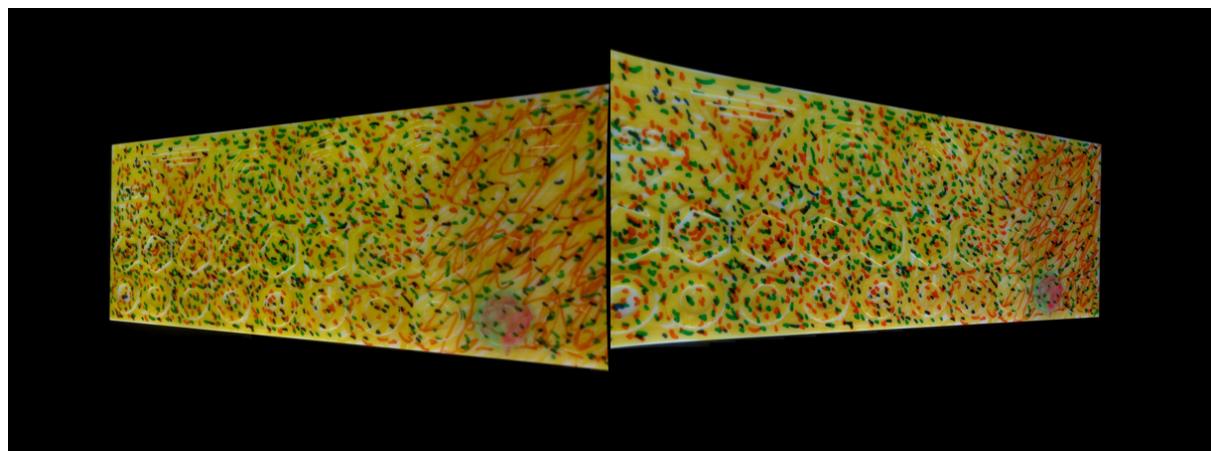


Figure 4.4: Montage of Left and Right Undistorted Images

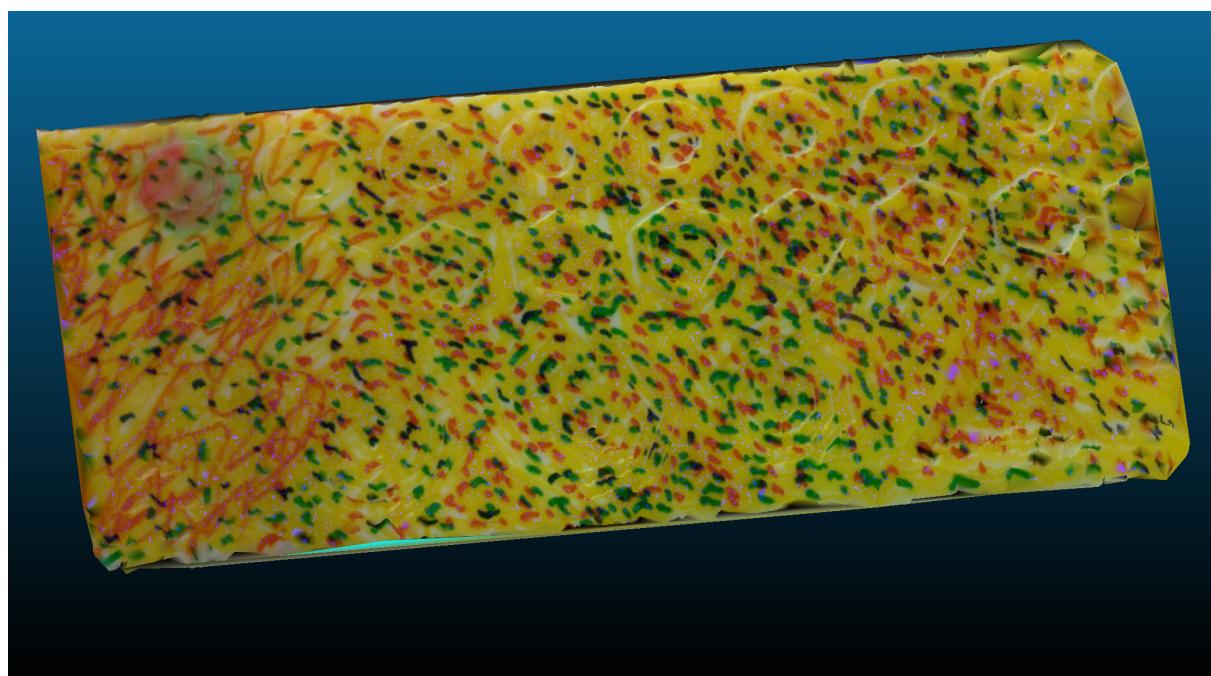


Figure 4.5: Mesh of the Yellow Pattern

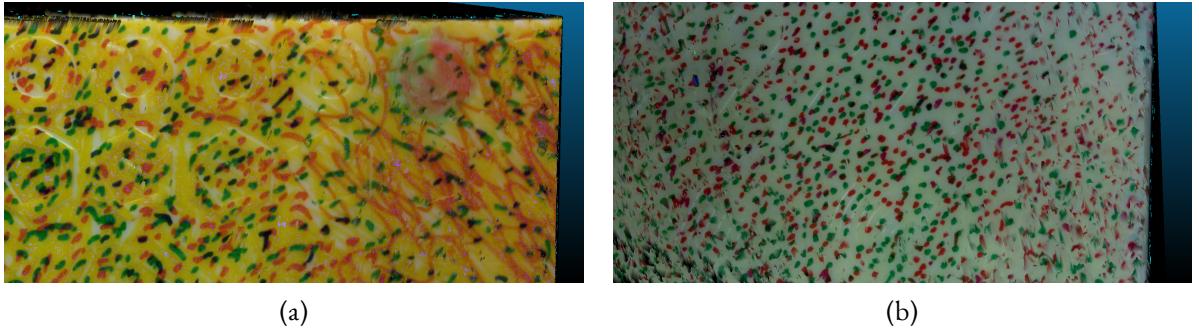


Figure 4.6: Regions of Interest of the Mesh

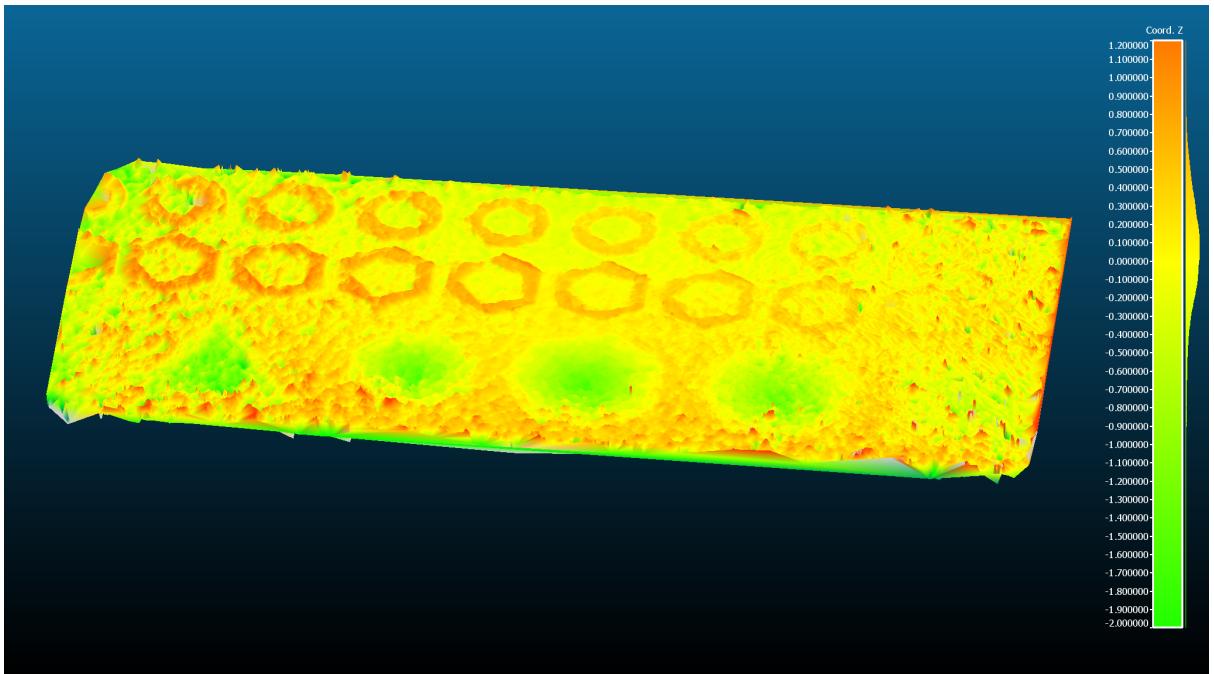


Figure 4.7: Height Map of the Reconstructed Workpiece

## 4.4 Discussions

This section discusses the results of the experiments, as reported in the previous section and their limitations.

### 4.4.1 Interpretation of the Results

Table 3.1 shows the estimated stereo parameters and compares it with actual values. It is seen that estimated and actual parameters for both experiments are close to actual parameters, and mean re-projection errors are sub-pixel.

The fig. 4.6 shows the region with  $0.1\text{ mm}$ – $0.7\text{ mm}$  embossed surface on the workpiece. It can be seen that this surface is well reconstructed. It is also seen that the workpiece with a yellow pattern is better reconstructed, mainly due to, better lightning conditions. During the experimentation process, it is found that the imaging quality is more dependent on lighting conditions than expected. It is also seen that the

central region of the rectified image is better reconstructed than the area at the end. This is expected because of the consequence of wide baseline and also evident from the fig. 4.4.

The different height levels are also clearly seen in the height map shown in fig. 4.7 (flipped). In this figure, hot colours are above the plane and cold colours are below. In this case, mesh is transformed such that fitted plane lies at  $z = 0$ . Further, there is a presence of spurious matches due to the absence of robust and statistical outlier detector and presence of mesh holes, which largely affect the quality of mesh.

From the results obtained in table 4.1 and by Grimm (2020), it can be concluded that the stereo camera system is able to differentiate between the distances in the order of 0.25 mm with limitations listed in the following section.

#### 4.4.2 Limitations and Recommendations

These experiments do not definitely conclude the usability of the camera on the human skin. For this purpose, experiments have to be conducted on the diseased human skin along with the investigation of different projection patterns apart from the Perlin noise. Currently, from the hardware perspective, the baseline and camera rotations are fixed, which can be made variable to obtain fine adjustment of the camera geometry as per the change in focal length due to auto-focus.

Further, it is also advised to collect ground truth data of the workpiece using sophisticated systems such as laser scanner to compare obtained results accurately. Results obtained in this document and by Grimm (2020) are subject to the accuracy of the 3D printer as no ground truth is available.

Further, to reduce the number of spurious matches and outliers algorithms such as RANSAC or SOR should be used. Reduction of spurious matches will significantly improve the quality of 3D reconstruction and thereby the mesh. Moreover, for improving quality of reconstruction Visual SLAM algorithms should also be considered as described in Part II.

From the usability perspective, the camera calibration and point cloud generation processes should be migrated to OpenCV. This is mainly due to the difference between coordinate transform mechanisms in OpenCV and MATLAB, which requires pre-processing of the matrices before use.

Pilot studies of using photometric stereo by means of projecting structured light are also of further interest.

## Part II

### *3D Reconstruction using Visual SLAM*



# Chapter 5

## Background

The purpose of this chapter is to provide background information and the formulation of the *error-state extended Kalman filtering* and *SLAM problem*. The theory developed in this chapter shall be used for the implementation.

### 5.1 Error-State Extended Kalman Filter

In the case of linear and dynamic process and measurement models with white Gaussian noise, the Kalman Filtering is optimal estimator which minimizes the mean square error. But, in real-world situations, such scenarios rarely occur. In the majority of scenarios, either the process or measurement model (or both) are nonlinear. In such cases, a class of suboptimal Kalman filter implementations called extended Kalman filters (EKF) are used. An alternative approach, called error state approach, for a certain class of problems where the error in the states is estimated using a Kalman filter, rather than the state itself. This approach derives the error state dynamics, via the perturbation of the nonlinear plant, lends itself to optimal updates in the error states and optimal prediction and updates in the error state covariance. This is because the error state dynamics are linear, thereby satisfying a condition for optimal Kalman filtering ([Madhyastha, Ravindra, Mallikarjunan, & Goyal, 2011](#)). The most remarkable properties of this filter are ([Solà, 2017](#)):

1. The orientation error-state is minimal.
2. The error-state always operate close to the origin, hence distant from possible parameter singularities, gimbal locks, thereby providing guarantee of linearisation all the time.
3. The error-state is always small, meaning that all second-order products are negligible.
4. The error dynamics are slow because all the large-signal dynamics have been captured by the nominal-state.

#### 5.1.1 Formulation

As mentioned in previous section, the ES-EKF holds the advantage that the actual Kalman filter is only applied to signals with small magnitude. This section quickly guides the reader through the formulation of the concept of Error State Kalman Filter by means for introducing the *error state* and the *nominal state*. For detailed explanation on this formulation refer: [Madhyastha et al. \(2011\)](#); [Solà \(2017\)](#); [van der Heijden \(2020\)](#).

Let the model of dynamic system be described by,

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{w}(t) \quad (5.1)$$

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}(t)) + \mathbf{v}(t) \quad (5.2)$$

Equation (5.1) describes the system equation indicating how true state at time  $t$  is converted to the true state at time  $t_1$ . The process noise  $\mathbf{w}(t)$  induces the uncertainty in the prediction  $\hat{\mathbf{x}}(t+1) = \mathbf{f}(t)$ , even if  $\mathbf{x}(t)$  is fully known.

Similarly, eqn. (5.2) describes the conversion of the true state  $\mathbf{x}(t)$  to the observed measurement  $\mathbf{z}(t)$ . The measurement noise  $\mathbf{v}(t)$  induces the uncertainty in the prediction  $\hat{\mathbf{z}}(t) = \mathbf{h}(t)$ , even if  $\mathbf{x}(t)$  is fully known.

The kinematics of ES-EKF splits the optimal estimate  $\hat{\mathbf{x}}(t | t)$  in two parts: the nominal state  $\check{\mathbf{x}}(t)$  and the estimated error state  $\hat{\delta\mathbf{x}}(t | t)$ . Hence,

$$\hat{\mathbf{x}}(t | t) = \check{\mathbf{x}}(t) + \hat{\delta\mathbf{x}}(t | t) \quad (5.3)$$

The nominal state vector takes care of large magnitudes of the state, whereas the error state represents the smaller deviations between optimal estimate and nominal state. The difference between the true, unknown state  $\mathbf{x}(t)$  and the nominal state is  $\delta\mathbf{x} = \mathbf{x} - \check{\mathbf{x}}$ .  $\hat{\delta\mathbf{x}}$  is the estimate of true, but unknown error in  $\delta\mathbf{x}$ .

Using truncated Taylor series expansion and rejecting higher order term, linearised system and measurement equations can be written as  $\mathbf{f}(\mathbf{x}(t)) \approx \mathbf{f}(\check{\mathbf{x}}(t)) + \mathbf{F}\hat{\delta\mathbf{x}}(t)$  and  $\mathbf{h}(\mathbf{x}(t)) \approx \mathbf{h}(\check{\mathbf{x}}(t)) + \mathbf{H}\hat{\delta\mathbf{x}}(t)$  respectively.  $\mathbf{F}$  and  $\mathbf{H}$  are the respective Jacobian matrices. Now, system equation can be written as,

$$\begin{aligned} \mathbf{x}(t+1) &\approx \mathbf{f}(\check{\mathbf{x}}(t)) + \mathbf{F}\hat{\delta\mathbf{x}}(t) + \mathbf{w}(t) \\ \mathbf{z}(t) &= \mathbf{h}(\check{\mathbf{x}}(t)) + \mathbf{H}\hat{\delta\mathbf{x}}(t) + \mathbf{v}(t) \end{aligned} \quad (5.4)$$

Using above formulation prediction and measurement equations can be written as,

$$\begin{aligned} \check{\mathbf{x}}(t+1) &= \mathbf{f}(\check{\mathbf{x}}(t)) \\ \hat{\delta\mathbf{x}}(t+1) &\approx \mathbf{F}\hat{\delta\mathbf{x}}(t) + \mathbf{w}(t) \\ \hat{\delta\mathbf{z}}(t) &= \mathbf{z}(t) - \mathbf{h}(\check{\mathbf{x}}(t)) = \mathbf{H}\hat{\delta\mathbf{x}}(t) + \mathbf{v}(t) \end{aligned} \quad (5.5)$$

The vector  $\hat{\delta\mathbf{x}}$  is the *state vector*, and  $\hat{\delta\mathbf{z}}$  is corresponding measurement vector. Equations for error state (eqn. (5.5)) are linear and Kalman filtering can be applied.

---

**Algorithm 5.1:** Error-State Kalman Filter; adapted from ([van der Heijden, 2020](#))

---

```

/* update or correct */
1  $\hat{\delta\mathbf{z}}(i) = \mathbf{z}(t) - \mathbf{h}(\check{\mathbf{x}}(t))$  // error state measurement vector
2  $\mathbf{H} = \frac{\partial}{\partial \mathbf{x}^T} \mathbf{h}(\mathbf{x}) \Big|_{\check{\mathbf{x}}(t)}$  // jacobian matrix
3  $\mathbf{S} = \mathbf{H}\mathbf{C}(t | t-1)\mathbf{H}^T + \mathbf{C}_v$  // covariance of innovation
4  $\mathbf{K} = \mathbf{C}(t | t-1)\mathbf{H}^T \mathbf{S}^{-1}$  // kalman gain
5  $\hat{\delta\mathbf{z}}(i) = \mathbf{H}\hat{\delta\mathbf{x}}(t | t-1)$  // predict measurements
6  $\hat{\delta\mathbf{x}}(t | t) = \hat{\delta\mathbf{x}}(t | t-1) + \mathbf{K}(\hat{\delta\mathbf{z}}(i) - \hat{\mathbf{z}}(i))$  // update estimated error state
7  $\mathbf{C}(t | t) = \mathbf{C}(t | t-1) - \mathbf{K}\mathbf{S}\mathbf{K}^T$  // covariance of estimated error state
8  $\check{\mathbf{x}}(t | t) = \check{\mathbf{x}}(t | t-1) \oplus \hat{\delta\mathbf{x}}(t | t)$  // nominal state correction
/* predict */
9  $\mathbf{F} = \frac{\partial}{\partial \mathbf{x}^T} \mathbf{f}(\mathbf{x}) \Big|_{\check{\mathbf{x}}(t | t)}$  // jacobian matrix
10  $\hat{\delta\mathbf{x}}(t+1 | t) = 0$  // reset error state
11  $\mathbf{C}(t+1 | t) = \mathbf{F}\mathbf{C}(t | t)\mathbf{F}^T + \mathbf{C}_w$  // covariance of predicted error state
12  $\check{\mathbf{x}}(t+1 | t) = \mathbf{f}(\check{\mathbf{x}}(t | t))$  // nominal state prediction

```

---

### 5.1.2 The Algorithm

Algorithm 5.1 presents the a single iteration of ES-EKF. Once the nominal state is corrected with the error state (line 8) the nominal state  $\hat{\mathbf{x}}(t | t)$  is unbiased with the zero mean error and will be zero. Hence, after embedding estimated error state in the nominal state, estimated error state becomes zero, making line 5 and line 10 obsolete. Further, this observation also simplifies the line 6 to  $\hat{\delta\mathbf{x}}(t | t) = \mathbf{K}\delta\mathbf{z}(i)$ .

The  $\oplus$  operator in the line 8 represents *generalized addition* as quaternions can not be simply added and eqn. (2.4) should be followed.

## 5.2 Simultaneous Localization and Mapping

The Simultaneous Localization and Mapping problem consists of three basic operations, reiterated at each step, viz., movement of robot, discovery of interesting features in the environment, called *landmarks*, and re-observation of these landmarks (Solà, 2014). Landmarks should be re-observable, distinguishable, stationary, and available in plenty (Riisgaard & Blas, 2003).

### 5.2.1 Basic Operations

**The Robot Movement:** This causes robot to explore the scene and either discover or revisit the landmarks. The uncertainty in the localization increasing due to unavoidable noise, called the *process noise*. This movement can be defined by a mathematical model known as the *motion model*, hereafter, this model is consistently referred as the *system model*.

**Discovery of New Landmarks:** These features must be added to the map. But due to noise in sensor measurements, called *measurement noise*, the location of the landmarks is also uncertain. The *inverse observation model*, is used to determine the position of the landmarks from the data obtained by the sensors. These landmarks are further stored in the landmark pool with their identifiable signature, which can be used for re-identification and matching of the landmarks at later stages.

**Re-observation of Old Landmarks:** Upon re-observing the old landmarks the information obtained is used to correct the both self-localization of and the localization of the landmarks. The *observation model*, hereafter, referred as the *measurement model* is used to predict the values of measurement from the predicted landmark location. The re-observation of landmarks gives rise to the problem of *data association*.

The algorithm described in the previous section is used as an estimator for the proper propagation of the uncertainties at each time instant. Further, the SLAM also needs to chain all these operations together and log data properly (we call it the *bookkeeping module*) to make appropriate decisions.

### 5.2.2 The Data Association Problem

Assume, at time instant  $t$ , a new scan is acquired and landmarks with their signature. These landmarks are matched with all the previously stored landmarks in the pool, so that they can me associated with their match. Hence, the problem of data association is that of matching observed landmarks from different scans with each other. This matching is done using a *data association policy* which is specific to the application area (Riisgaard & Blas, 2003). While associating currently detected landmarks with pool following issues can arise:

- **Landmark might not be re-observed at every time step:** This could be because bad landmark extraction or association policy or simply landmark is not visible at current time instant.

- **Landmark may not be seen ever again:** This is mainly due to bad landmark extraction and association policy, assuming loop closure takes place. Such landmarks should be deleted when pool is exhausted.
- **Wrong association to previously seen landmark:** This issue has devastating effect of the generalization ability of the SLAM. A good *landmark forking policy* may help to detect and overcome this situation, at subsequent time instants this landmark may be matched to multiple previously seen landmark.

### 5.2.3 The Matrices

As SLAM algorithm progresses, the size of various matrices involved in the algorithm, such as state vector, landmark pool, Jacobian matrices, covariance matrices changes. The bookkeeping module takes care of maintaining proper dimensionality such matrices. This subsection gives a short description of such matrices. This section also standardizes the notions from the perspective visual ES-EKF SLAM for the application mentioned in section 1.1. For in depth description and derivation a report by [Riisgaard and Blas \(2003\)](#) and [van der Heijden \(2020\)](#) should be referred.

#### The State Vector

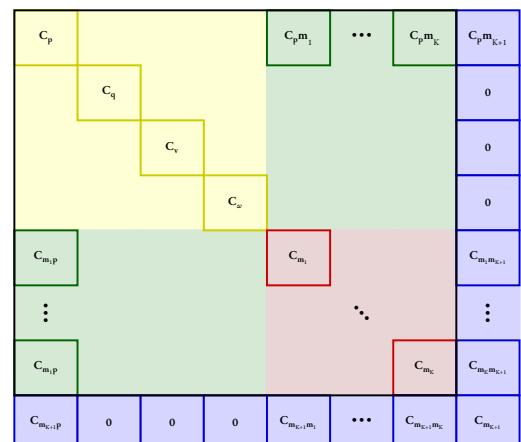
The state vector with its covariance matrix (defined below) is one of the most important matrix in construction of the SLAM framework. It contains the pose of the robot  ${}^A\mathbf{p}(t)$ ,  ${}^A\mathbf{q}(t)$ , linear and angular velocity of the robot  ${}^B\mathbf{v}_{AB}(t)$ ,  ${}^B\boldsymbol{\omega}_{AB}(t)$  and position of the all seen landmarks  $\mathbf{m}_{1:K}$ , subject to their deletion. Hence, dimensions of state vector is  $13 + 3K$ . As soon as new landmark is seen, this landmark is added to the state vector. The process of addition of new landmark is called *augmentation*.

In concern to ES-EKF estimator, as defined in section 5.1.1, this true state vector is composed of two parts a nominal state and error state. It is important to note that error state vector does not contain the scalar part of the quaternion while defining the pose owing to the fact that, scalar part  $\delta q_w$  is always 1 and excluding this from state vector make system equation for quaternion linear. Hence, the dimensions of error state vector is  $12 + 3K$ .

$$\mathbf{x}(t) := \begin{bmatrix} {}^A\mathbf{p}(t) \\ {}^A\mathbf{q}(t) \\ {}^B\mathbf{v}_{AB}(t) \\ {}^B\boldsymbol{\omega}_{AB}(t) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_k \\ \vdots \\ \mathbf{m}_K \end{bmatrix}$$

#### The Covariance Matrix

The covariance matrix  $\mathbf{C}$  is very central to understanding of the system as it encapsulates the uncertainties in the states and landmark positions. This matrix contains covariance on states (**yellow**), covariance between states and landmark positions (**green**), and covariance between landmarks (**red**). The dimensions of this matrix are  $(13 + 3K) \times (13 + 3K)$ . Covariance matrix of the error state vector is smaller by 1 along both dimensions and hence the Kalman Gain Matrix. When state vector is augmented, the covariance matrix also has to be augmented accordingly. Say if  $K$  landmarks are already visible, when new landmark is found, the covariance matrix is augmented as shown by **blue** blocks. If the predicted state is augmented, a possible assumption about the initial position of the new landmark is that it is at a position in front of the device in a direction that is indicated by the measurement  $\mathbf{z}_{K+1}(t)$ , and at a fixed distance, say  $D$ . As the knowledge about this distance  $D$  is poor, a large uncertainty  $\sigma_D$  in this distance is assumed. The operations required to augment the covariance matrix are



given below.

$$\mathbf{C}_{\mathbf{m}_{K+1}} = \mathbf{C}_p + \sigma_D^2 \mathbf{I} \quad (5.6)$$

$$\mathbf{C}_{\mathbf{p}\mathbf{m}_{K+1}} = \mathbf{C}_p \quad (5.7)$$

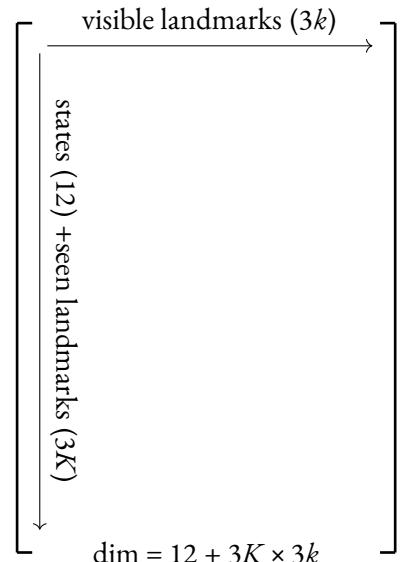
$$\mathbf{C}_{\mathbf{m}_k \mathbf{m}_{K+1}} = \mathbf{C}_{\mathbf{m}_k \mathbf{p}} \quad (5.8)$$

If any landmark from the pool is deleted corresponding entries in the covariance matrix are also deleted.

### The Kalman Gain

The Kalman gain matrix  $\mathbf{K}$  is the relative weight given to the measurements and current state estimate. If measurement are quite accurate, that is, we put much trust in the sensory data than the system model, then the magnitude of Kalman Gain is small and measurements highly influence the next state estimate. If this gain is quite high, implies the sensory information is quite poor and system model dominates the estimation. In case of ES-EKF SLAM, this Kalman gain matrix concerns with the error state.

While dealing with SLAM, the Kalman gain matrix may have different dimensions each time step. The dimension along the row, i.e., columns describe how much innovation can be gained for a particular parameter. Number of columns depends upon number of currently visible landmarks and number of rows depends upon the dimension of the state vector and number of landmarks in the pool.



### The Jacobian

Jacobian of the system function or measurement function explain the dynamics of the first order error propagation. The Jacobain of system function evaluated at  $\hat{\mathbf{x}}(t | t)$  to provide a way to propagate the uncertainty in the estimate  $\hat{\mathbf{x}}(t | t)$  to the estimation of  $\hat{\mathbf{x}}(t + 1 | t)$ .

Similarly, Jacobian of measurement function explains the propagation of prediction error of the estimate to the predicted measurement.

### The Process Noise

To incorporate uncertainties in the model, due to lack of knowledge etc., process noise is introduced in the state equation, so that the magnitude of uncertainty can be quantified. It is assumed that this noise is white, zero-mean and Gaussian. In this document, the covariance matrix of the process noise is denoted by  $\mathbf{C}_w$ .

### The Measurement Noise

The measurement device or sensor introduces an uncertainty due to variety or reasons, such as readout, quantization, environmental effects etc. These noises are incorporate in the measurement model as measurement noise. It is again assumed that this noise is white, zero mean and Gaussian. In this document, the covariance matrix of the measurement noise is denoted by  $\mathbf{C}_v$ .

## 5.2.4 Loop Closure

When robot revisits the previously mapped area it is termed as loop closure. The uncertainties in robot and landmark estimates are then reduced if the loop is correctly closed.

## 5.3 Observability Analysis

Observability is a notion that plays an important role in filtering and reconstruction of the states from inputs and outputs. To put more formally, observability is concerned with the problem of determining the state of a dynamic system from observations of the output and control vectors in a finite number of sampling periods (Ogata, 1995, 2010). This concept was first introduced by Kalman.

Following section discusses about the complete observability in deterministic discrete time systems. As already established that, error-state kinematics are linear in nature, this observability analysis can be used to assess the complete observability of the error-state system.

### 5.3.1 Complete Observability in Discrete Time Systems

Let discrete time system be defined as

$$\mathbf{x}((k+1)T) = \mathbf{A}(kT)\mathbf{x}(kT) \quad (5.9)$$

$$\mathbf{y}(kT) = \mathbf{C}(kT)\mathbf{x}(kT) \quad (5.10)$$

where,

$\mathbf{x}(kT)$  = state vector ( $n$ -vector) at  $k^{\text{th}}$  sampling instant

$\mathbf{y}(kT)$  = output/measurement vector ( $m$ -vector) at  $k^{\text{th}}$  sampling instant

$\mathbf{A}(kT)$  =  $n \times n$  matrix, also known as *system matrix* at  $k^{\text{th}}$  sampling instant

$\mathbf{C}(kT)$  =  $m \times n$  matrix, also known as *output/measurement matrix* at  $k^{\text{th}}$  sampling instant

As reported by Bar-Shalom and Li (1993); Van Der Heijden, Duin, De Ridder, and Tax (2005b), *Observability Gramian*,  $\mathcal{G}$ , can be calculated for evaluation of complete observability. The output/measurement equation (5.10), is of type  $\mathbf{y} = \mathbf{Cx}$ , for which least squares estimate is given as (Bar-Shalom & Li, 1993; Van Der Heijden, Duin, De Ridder, & Tax, 2005a),

$$\hat{\mathbf{x}} = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{y} \quad (5.11)$$

The solution of above equation exists if and only if the  $(\mathbf{H}^\top \mathbf{C})^{-1}$  exists, that is, the rank of  $(\mathbf{C}^\top \mathbf{C})$  is equal to the dimension  $n$  of the state vector. Equivalent conditions are that the matrix  $(\mathbf{C}^\top \mathbf{C})$  is a positive definite, or all eigenvalues are positive. The observability Gramian can be defined as,

$$\mathcal{G} = \mathbf{C}^\top(kT) \mathbf{C}(kT) + \sum_{j=1}^n \left( \mathbf{C}(kT + nT) \prod_{l=0}^{j-1} \mathbf{A}(kT + lT) \right)^\top \left( \mathbf{C}(kT + nT) \prod_{l=0}^{j-1} \mathbf{A}(kT + lT) \right) \quad (5.12)$$

### Linear Time Invariant Systems

For system to be completely observable, given the output  $\mathbf{y}(kT)$  over a finite number of sampling periods, it is possible to determine the initial state vector  $\mathbf{x}(0)$ . Also, for LTI systems, system and output/measurement matrices do not depend upon time. Hence, the solution of the system defined in eqs. (5.9) and (5.10)

can be written as,

$$\mathbf{x}(kT) = \mathbf{A}^k \mathbf{x}(0) \quad (5.13)$$

$$\mathbf{y}(kT) = \mathbf{C}\mathbf{A}^k \mathbf{x}(0) \quad (5.14)$$

The rank of  $\mathcal{G}$  is equal to dimension of the state vector and is positive definite at the same time. Hence, eqn. (5.14) evaluates to be,

$$\mathcal{G} = \sum_{j=0}^n (\mathbf{C}\mathbf{A}^j)^T (\mathbf{C}\mathbf{A}^j)$$

For stable systems, condition  $n \rightarrow \infty$  can be set to evaluate observability. Hence, necessary and sufficient conditions are  $\mathcal{G} > 0$  and  $\text{rank}(\mathcal{G}) = n$ .

Alternatively, for complete observability, given  $\mathbf{y}(0)$ ,  $\mathbf{y}(T)$ ,  $\mathbf{y}(2T)$ , ..., it should be possible to determine initial values of all the state variables, i.e., it should be possible to determine  $\mathbf{x}(0) = [x_1(0) \ x_2(0) \ x_3(0) \ \dots \ x_n(0)]$ . To determine these  $n$  unknowns, we need a system of  $n$  linear equations. This  $n \times m$  system of simultaneous equations is mathematically written as,

$$\begin{aligned} \mathbf{y}(0) &= \mathbf{Cx}(0) \\ \mathbf{y}(T) &= \mathbf{CAx}(0) \\ &\vdots \\ \mathbf{y}((n-1)T) &= \mathbf{CA}^{n-1}\mathbf{x}(0) \end{aligned}$$

To obtain the unique set of solutions from this system of simultaneous equations, all  $n$  equations should be linearly independent. In vectorized notation this condition requires that  $nm \times n$  dimensional *observability matrix*,

$$\mathcal{O} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \\ \mathbf{CA}^{n-1} \end{bmatrix} \quad (5.15)$$

should be of rank  $n$ . Hence, the necessary and sufficient condition for the system to be completely observable is  $\text{rank}(\mathcal{O}) = n$ .



# Chapter 6

## Design and Implementation

This chapter presents the *system model* and the *measurement model* of the stereo-camera system used by the ES-EKF SLAM algorithm. Further, this chapter also discusses the implementation of ES-EKF SLAM algorithm by highlighting the data simulation and collection strategy.

### 6.1 Kinematics of ES-EKF SLAM

Following sections report the state vector, system and measurement functions of the Stereo Camera System used for ES-EKF presented in the algorithm 5.1. For complete derivation, a technical report by [van der Heijden \(2020\)](#) should be referred.

#### 6.1.1 The State Vector

The state vector  $\mathbf{x}(t)$  contains the current pose of the moving stereo camera and velocities for the prediction of the next pose. Hence, the state vector comprises of 3D position  ${}^{\text{wcs}}\mathbf{p}(t)$ , 3D orientation  ${}^{\text{wcs}}\mathbf{q}(t)$ , 3D linear velocity  ${}^{\text{ccs}}\mathbf{v}_{\text{wcs,cam}}(t)$  and angular velocity  ${}^{\text{ccs}}\boldsymbol{\omega}_{\text{wcs,cam}}(t)$ .

Hence, the state vector is a  $13 + 3K$  dimensional vector with  $12 + 3K$  degrees of freedom.

$$\mathbf{x}(t) = \begin{bmatrix} {}^{\text{wcs}}\mathbf{p}(t) \\ {}^{\text{wcs}}\mathbf{q}(t) \\ {}^{\text{ccs}}\mathbf{v}_{\text{wcs,cam}}(t) \\ {}^{\text{ccs}}\boldsymbol{\omega}_{\text{wcs,cam}}(t) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_K \end{bmatrix} \in \mathbb{R}^{13+3K} \quad (6.1)$$

From section 5.2.3, a  $12 + 3K$  dimensional error state vector can be written by removing the scalar part of the  ${}^{\text{wcs}}\mathbf{q}(t)$ .

### 6.1.2 The System Function

As per [van der Heijden \(2019c\)](#), the system can be modelled as,

$$\begin{aligned} {}^{\text{wcs}}\mathbf{p}(t+1) &= {}^{\text{wcs}}\mathbf{p}(t) + {}^{\text{wcs}}\mathbf{R}_{\text{ccs}} \mathbf{v}_{\text{wcs,cam}}(t) \Delta \\ {}^{\text{wcs}}\mathbf{q}(t+1) &= e^{(\mathbf{W}\Delta)} {}^{\text{wcs}}\mathbf{q}(t) \\ {}^{\text{ccs}}\mathbf{v}_{\text{wcs,cam}}(t+1) &= \Gamma^{\text{ccs}} \mathbf{v}_{\text{wcs,cam}}(t) + \mathbf{w}_1(t) \\ {}^{\text{ccs}}\boldsymbol{\omega}_{\text{wcs,cam}}(t+1) &= \Lambda^{\text{ccs}} \mathbf{v}_{\text{wcs,cam}}(t) + \mathbf{w}_2(t) \end{aligned} \quad (6.2)$$

where, diagonal matrices,  $\Gamma$  and  $\Lambda$  are linear and angular velocity forgetting factors, respectively, holding values between 0 and 1. The vectors  $\mathbf{w}_1(t)$  and  $\mathbf{w}_2(t)$  are assumed to be white, zero mean, and Gaussian with  $3 \times 3$  diagonal covariance matrices  $\mathbf{C}_1$  and  $\mathbf{C}_2$ , represent the unpredictable part of the system.  $\delta$  is a sampling period, and  ${}^{\text{wcs}}\mathbf{R}_{\text{ccs}}$  and  $\mathbf{W}$  are orientation-dependent matrices.

$$\mathbf{W} = \frac{1}{2} \begin{bmatrix} 0 & \omega_x & \omega_y & \omega_z \\ -\omega_x & 0 & \omega_z & -\omega_y \\ -\omega_y & -\omega_z & 0 & \omega_x \\ -\omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

Hence, the system model of ES-EKF SLAM in the vectorized form in accordance with eqn. (5.1) can be written as,

$$\mathbf{x}(t+1) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 4} & {}^{\text{wcs}}\mathbf{R}_{\text{ccs}} \Delta & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{4 \times 3} & e^{\mathbf{W}\Delta} & \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 3} & \mathbf{0}_{4 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 4} & \Gamma & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 4} & \mathbf{0}_{3 \times 3} & \Lambda & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 4} & \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{I}_{3K \times 3K} \end{bmatrix} \begin{bmatrix} {}^{\text{wcs}}\mathbf{p}(t) \\ {}^{\text{wcs}}\mathbf{q}(t) \\ {}^{\text{ccs}}\mathbf{v}_{\text{wcs,cam}}(t) \\ {}^{\text{ccs}}\boldsymbol{\omega}_{\text{wcs,cam}}(t) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_K \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{4 \times 1} \\ \mathbf{w}_1(t) \\ \mathbf{w}_2(t) \\ \mathbf{0}_{3K \times 1} \end{bmatrix} \quad (6.3)$$

From [van der Heijden \(2020\)](#), the error state function,

$$\begin{aligned} \delta\mathbf{x}(t+1) &= \delta\mathbf{f}(\delta\mathbf{x}(t)) + \mathbf{w}(t) \\ &= \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & {}^{\text{wcs}}\mathbf{R}_{\text{ccs}} \Delta & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{1}{2}\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Gamma & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Lambda & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{I}_{3K \times 3K} \end{bmatrix} \begin{bmatrix} {}^{\text{wcs}}\delta\mathbf{p}(t) \\ {}^{\text{wcs}}\delta\mathbf{q}(t) \\ {}^{\text{ccs}}\delta\mathbf{v}_{\text{wcs,cam}}(t) \\ {}^{\text{ccs}}\delta\boldsymbol{\omega}_{\text{wcs,cam}}(t) \\ \delta\mathbf{m}_1 \\ \vdots \\ \delta\mathbf{m}_K \end{bmatrix} + \mathbf{w}(t) \end{aligned} \quad (6.4)$$

and the Jacobian matrix

$$\mathbf{F}(\hat{\mathbf{x}}(t \mid t)) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{P}_\theta & {}^{\text{wcs}}\mathbf{R}_{\text{ccs}} \Delta & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \frac{1}{2}\mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Gamma & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \Lambda & \mathbf{0}_{3 \times 3K} \\ \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{0}_{3K \times 3} & \mathbf{I}_{3K \times 3K} \end{bmatrix} \quad (6.5)$$

### 6.1.3 The Measurement Function

Stereo camera systems are capable of reconstructing 3D position of landmarks from two or more camera images in its own coordinate system, here denoted by  $\text{ccs}$ . This process is explained in section 2.8 and algorithm 2.1.

Let this 3D position be denoted by  ${}^{ccs}\mathbf{z}_k(t)$  with its corresponding covariance matrix  ${}^{ccs}\mathbf{C}_k(t)$ .

$${}^{ccs}\mathbf{z}_k(t) = {}^{wcs}\mathbf{R}_{ccs}^T ({}^{wcs}\mathbf{m}_k - {}^{wcs}\mathbf{p}) + \mathbf{v}_k(t) \quad (6.6)$$

In the error model, the measurement that is used is  $\delta\mathbf{z}_k(t) = \mathbf{z}(t) - \mathbf{h}(\check{\mathbf{x}}(t))$ , so that the model becomes  $\delta\mathbf{z}_k(t) = \mathbf{H}\delta\mathbf{x}(t) + \mathbf{v}(t)$ , where  $\mathbf{H}$  is the Jacobian matrix of measurement function  $\mathbf{h}(\cdot)$ . The following equations of measurement function and Jacobian matrix are derived in [van der Heijden \(2020\)](#)

$$\mathbf{h}_k(\mathbf{x}(t)) = \begin{bmatrix} -{}^{wcs}\mathbf{R}_{ccs}^T & \mathbf{0}_{3 \times (7+3k)} & {}^{wcs}\mathbf{R}_{ccs} & \mathbf{0}_{3 \times (K-k)} \end{bmatrix} \begin{bmatrix} {}^{wcs}\mathbf{p}(t) \\ {}^{wcs}\mathbf{q}(t) \\ {}^{ccs}\mathbf{v}_{wcs,cam}(t) \\ {}^{ccs}\boldsymbol{\omega}_{wcs,cam}(t) \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_k \\ \vdots \\ \mathbf{m}_K \end{bmatrix} \quad (6.7)$$

and

$$\mathbf{H}_k(\mathbf{x}(t)) = \begin{bmatrix} -{}^{wcs}\mathbf{R}_{ccs}^T & 2[\mathbf{h}_k(\mathbf{x}(t))]_x & \mathbf{0}_{3 \times (3+3k)} & {}^{wcs}\mathbf{R}_{ccs}^T & \mathbf{0}_{3 \times (K-k)} \end{bmatrix} \quad (6.8)$$

where,  $[\cdot]_x$  is a skew-symmetric cross-product operator.

## 6.2 Implementation

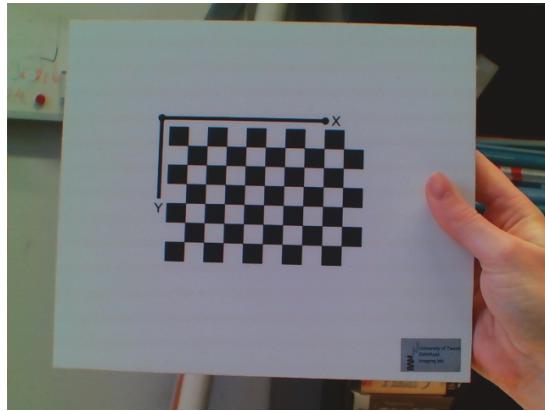
This section explains implementation of the ES-EKF SLAM algorithm from the data collection, visualization to getting it working. The stereocamera used for the implementation has following specifications:

1. focal length: 1.8 mm
2. pixel size: 1.12 mm
3. image size: 480 pixels  $\times$  640 pixels
4. baseline vector:  $x = 105$  mm,  $y = 0$  mm,  $z = 15$  mm

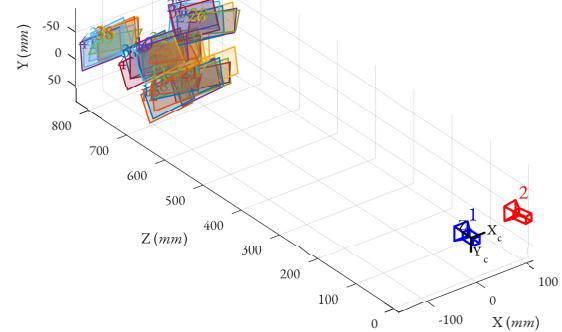
### 6.2.1 Camera Calibration

As previously shown, the field of view of our cameras is well below 95°, the *Stereo Camera Calibrator App* from MATLAB's Computer Vision Toolbox is used to calibrate the stereo camera system. A planar chequerboard pattern (each square 10 mm  $\times$  10 mm, fig. 6.1a) is used for calibration.

While calibrating system, several pictures of the board shown at different positions and orientations are captured. It is also made sure that the camera is filled as much as possible.



(a) Checker Board used for Calibration



(b) Visualization of Extrinsic Parameters

Figure 6.1: **Camera Calibration:** Chequerboard pattern and visualization of extrinsics after calibration using MATLAB Camera Calibrator Toolbox

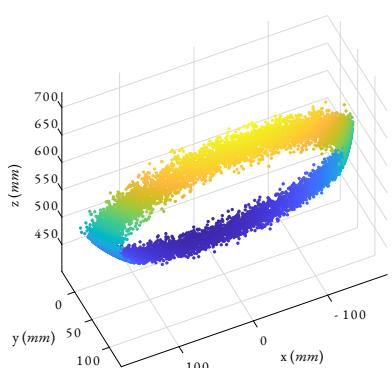
### 6.2.2 Choice of the Test Object

The choice of globe as a test object is a mere consequence of the COVID-19 country-wide restrictions. Further, a few more reasons behind selecting a globe as handy test object are:

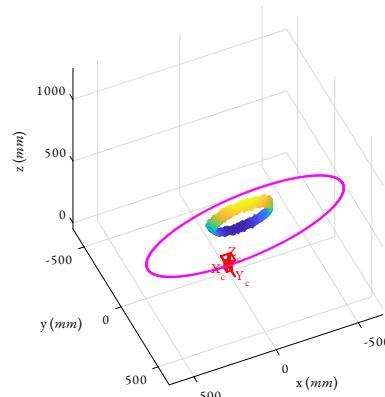
1. It is easy to rotate the globe with almost constant velocity
2. Geometry of sphere is well known, hence, it is easy to debug
3. It is rich in texture

### 6.2.3 Globe Simulator

For initial experimentation and inspecting a working of the SLAM, a simulator is designed which mimics the real-world data. This simulator mimics a camera, orbiting a globe while measuring 3D landmarks. To make this simulator more realistic and usable in this scenario, each landmark (on the surface of the globe) consists of a descriptor vector of length 64 (inspired from KAZE features). This descriptor shall be used in SLAM to for landmark matching. Following are the images of the simulated globe, camera trajectory. The fig. 6.2a shows the point cloud representation of the simulated globe and camera trajectory around it.



(a) Simulated Globe



(b) Camera Trajectory around the Globe

Figure 6.2: **Simulated Data Collection:** Simulated globe and movement of camera around it

#### 6.2.4 Real-World Data

As already mentioned, simulator mimics the real world; real data is collected by capturing the images of the rotating globe by the stereo camera. One such image is shown in the fig. 6.4. The important condition in ES-EKF is to have constant velocity. As, the globe is rotated manually, efforts have been taken to make sure the velocity is near-constant.

This data is further pre-processed to make it usable for ES-EKF SLAM algorithm. Following measures are taken in order:

1. undistort image using parameters estimated during camera calibration (section 6.2.1).
2. detect and extract KAZE features.
3. matching the features (fig. 6.5).
4. sorting these matched features according to their match strength and localization.
5. estimating the 3D position of these features from 2D image position using theory discussed in section 2.8 and algorithm 2.1



Figure 6.3: **Globe used for Real Data Collection:** image taken using cellphone camera



Figure 6.4: **Real Data:** Montage of the image captured

#### 6.2.5 The ES-EKF SLAM

The error-state extended Kalman filter is central to the SLAM algorithm used in this document. A MATLAB script is written in accordance with algorithm 5.1. While using this algorithm, it is important to note that a coordinate frame CCS is affixed to the stereocamera, moving with respect to a fixed global frame WCS. This means that the angular errors are relative to the WCS and not current body configuration. Hence, *global angular errors* are considered. The global definition of the angular error  $\delta q$  implies a generalized addition of quaternions on the left-hand side in contrast to the IMU driven (considering local angular errors) ES-EKF, which would multiply it on the right-hand side (Parwana & Kothari, 2017; Solà, 2017).

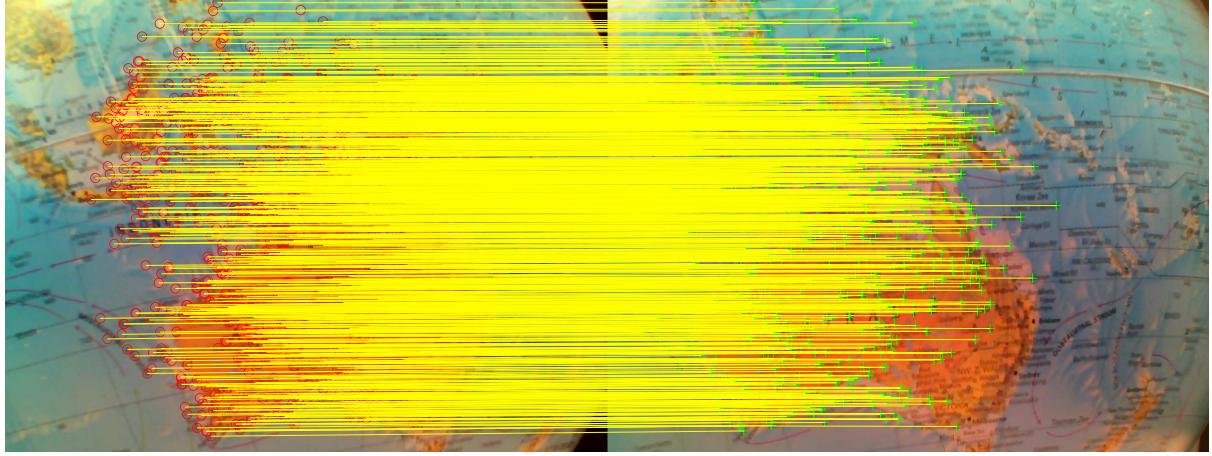


Figure 6.5: Matched Features using KAZE Descriptors on the Real Data

This ES-EKF Script is incorporated in the SLAM framework in accordance with the theory described in section 5.2. The actual plot for real data can not be drawn beforehand but should be estimated using the above developed SLAM framework.

### 6.2.6 Proposed Pipeline

Figure 6.6 shows the proposed pipeline for the execution of the ES-EKF SLAM algorithm. The top blocks are the sensory inputs to the algorithm which shall be optimally combined by means of the ES-EKF.

The stereo camera system is used to capture the image. The KAZE features with KAZE descriptors of length 64 are extracted from the left and right camera image. The correspondences of the 2D features in the stereo pair are used to extract 3D landmarks from the scene. Position of these landmarks in the camera coordinate system with their corresponding uncertainty and descriptor is stored.

During the SLAM algorithm, the stored descriptor is used for the association of landmarks with the previously seen landmarks in the pool. Thus, the camera pose in the global coordinate frame and relative camera motion in the camera coordinate frame is estimated. While testing the algorithm on simulated data, angular velocity, and linear velocity shall also be used for state correction. While testing on real data, other sensory inputs are not considered due to unavailability of the hardware at this stage.

### 6.2.7 The Landmark Pool

During runtime of the algorithm, all previously seen and new landmarks are stored in the landmark pool. Currently, the pool is characterized by:

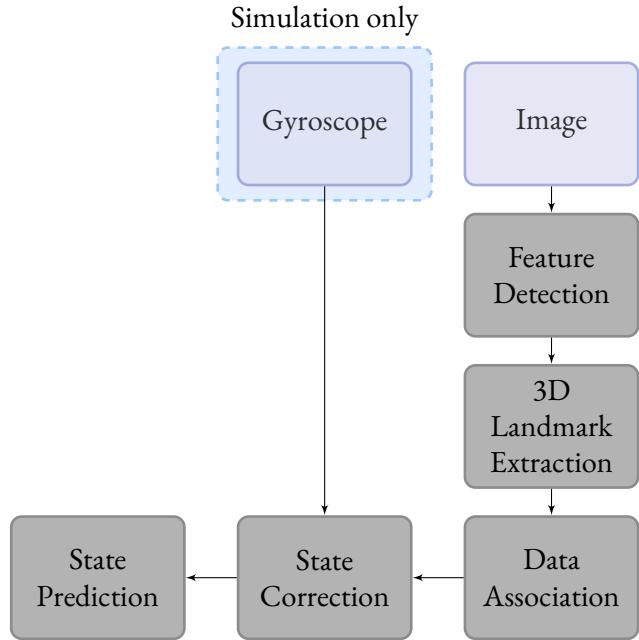


Figure 6.6: The Proposed SLAM Pipeline

- 1. current pool size
- 2. number of all seen landmarks
- 3. unique ID of landmarks
- 4. in-use status of the landmark
- 5. descriptors of the landmarks
- 6. match strength of the landmarks
- 7. maximum run-length of the landmarks
- 8. landmark index in the error-state vector
- 9. dimension of the error state vector
- 10. birth date of the landmarks

### Addition of the New Landmarks

Every newly seen landmark is added to the pool with its characteristic information as enumerated above. Upon addition of new landmark(s), augmentation of the state vector and its corresponding covariance matrix is augmented.

### Deletion of the Old Landmarks

Deletion of the old landmark takes place when either the pool has exhausted or landmark forking is detected. Upon pool exhaustion, landmarks must be strategically removed. Here, for the sake of getting algorithm working, two policies are implemented: *i) based on age, ii) based on run length.*

#### 6.2.8 Landmark Forking

The issue of landmark forking occurs at a data association stage when a currently visible landmark matches to multiple landmarks in the landmark pool. If landmark forking is detected, which is usually a case, the matchings should be resolved by a *landmark forking policy*. Two policies are implemented: *i) landmark with maximum run length is associated, ii) landmark which was seen first is associated*. If an erroneous association is made at this stage, it has a disastrous impact on the generalization capability of the algorithm. Such erroneous association is deleted from the pool.

The forgetting factors  $\Gamma$  and  $\Lambda$  are set as 1. This is because the velocity of the camera in the consecutive frame highly depends on the previous frame as the incremental and smooth change in camera position is assumed.



# Chapter 7

# Experiments

This chapter describes the experiments conducted with the system designed in the previous chapters. This chapter also elucidates the motivation behind the experiments, experimental protocol and methods and results obtained. Finally, the results obtained are discussed in the discussion section.

## 7.1 Motivation and Goals

To validate the working of ES-EKF as an estimator and its usage in the framework of visual SLAM *four* experiments are proposed. The motivation behind conducting these experiments is to validate the use of ES-EKF as an estimator for the visual SLAM framework discussed in chapters 5 and 6.

Hence, the following questions are of interest:

1. Is ES-EKF able to sufficiently estimate the trajectory of the camera when landmark position is already known in the world coordinate system?
2. Is ES-EKF SLAM able to sufficiently estimate the landmark location for 3D reconstruction and trajectory of the camera when landmark location is unknown in the world coordinate system?
3. Can covariance matrix be used to understand the error propagation and error dynamics intuitively?
4. Is it possible to comment on the ambiguity between translation along the  $x$ -axis and the rotation along the  $y$ -axis by means of the covariance matrix of estimation?
5. Does the addition of gyroscope in simulated data show any improvement in the accuracy of reconstruction?

The experiments proposed to answer these questions are described in the following section. Further, the observability analysis of each system is also of interest. The theory behind this is explained in section 5.3.

## 7.2 Methods

To answer questions asked in the previous section experiments are conducted on the simulated as well as the real data. The real data collection strategy is described in section 6.2.4. For the collection of the real data, a stereocamera which is already available at the department of Robotics and Mechatronics is used. For this purpose, MATLAB R2020a is used. It is explicitly mentioned that *Image Acquisition Toolbox* is necessary for operating the stereocamera system through MATLAB. In addition, for camera calibration, feature detection, etc., *Computer Vision Toolbox* is required on the top of the default installation.

Table 7.1 outlines the proposed experiments with their respective data collection strategies, evaluation metrics in the well tabulated manner.

Table 7.1: Proposed Experiments: Motivation and Methods

Expt. №	Description	Data Collection Strategy	Method	Plots of Interest	Evaluation Metric
1	ES-EKF only	Globe Simulator (section 6.2.3) shall be used to capture landmark locations in both WCS and CCS	Landmark positions in CCS and known transform ${}^{wcs}T_{ccs}$ shall be used to estimate camera pose and velocities	1. Pose Plot 2. Velocity Plot	-
2	ES-EKF SLAM without a gyroscope on simulated data	Globe Simulator (section 6.2.3) shall be used to capture landmark locations in both WCS and CCS	Only landmark positions in CCS shall be used. Transform shall be estimated during the each step to obtain reconstruction in WCS	1. Pose Plot 2. Velocity Plot	1. Cloud to Cloud distance 2. Fitting a sphere to point cloud
3	ES-EKF SLAM with a gyroscope on simulated data	Globe Simulator (section 6.2.3) shall be used to capture landmark locations in both WCS and CCS and camera motion in CCS	only landmark positions in CCS shall be used. Transform shall be estimated during the each step to obtain reconstruction in WCS. It is assumed that camera and gyroscope share the same coordinate frame and origin, i.e., ${}^{gcs}T_{ccs}$ is $\mathbf{I}$	1. Pose Plot 2. Velocity Plot 3. Comparison with expt. 2	1. Cloud to Cloud distance 2. Fitting a sphere to point cloud
4	ES-EKF SLAM on real data	Real data shall be used as explained in the section 6.2.4	Only landmark positions in CCS shall be used. Transform shall be estimated during the each step to obtain reconstruction in WCS	1. Pose Plot 2. Velocity Plot	Fitting a sphere to point cloud

The parameters set for the globe simulator and SLAM algorithm used are as follows:

- Radius of the globe: 200 mm
- Distance of the camera from the centre of the globe: 600 mm
- Total Number of landmarks on the globe: 10000 (assumption: uniformly distributed over the surface)
- Latitudinal spread of landmarks: 10 mm
- Axial tilt of the globe: 23 °
- Latitude to follow:  $-25^\circ$  (Commonwealth of Australia)
- Number of samples: 420
- Angular velocity:  $0.15 \text{ rad s}^{-1}$
- Number of landmarks increment: 100 or 50% upon pool exhaustion
- Maximum pool size: 1000

### 7.3 Results

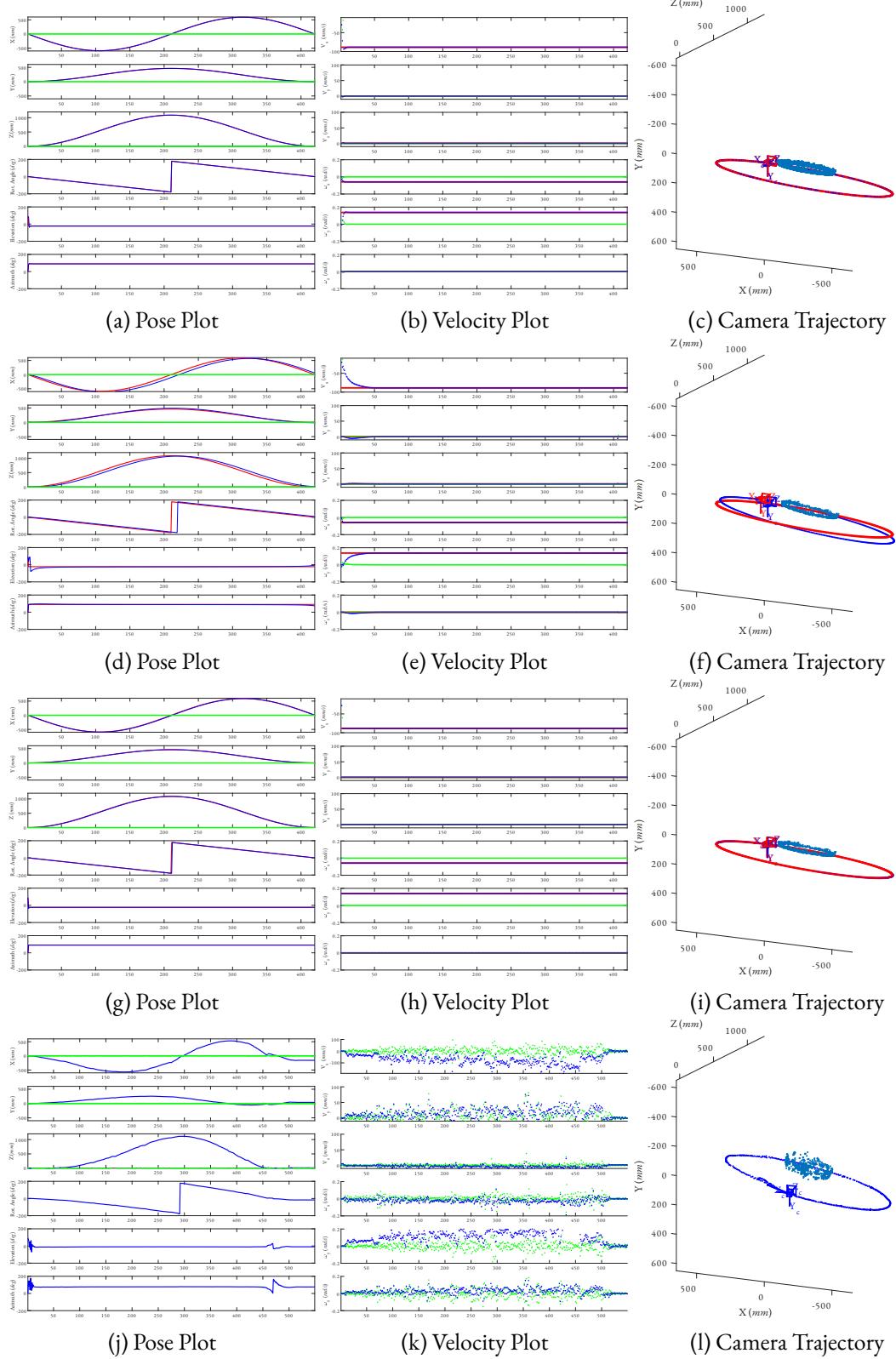


Figure 7.1: Results of the Experiments: (a)–(c): Experiment 1–ES-EKF only; (d)–(f): Experiment 2–ES-EKF SLAM without gyroscope; (g)–(i): Experiment 3–ES-EKF SLAM with gyroscope; (j)–(l): Experiment 4–ES-EKF SLAM on real data; Legend: reference, error state, estimated/nominal state

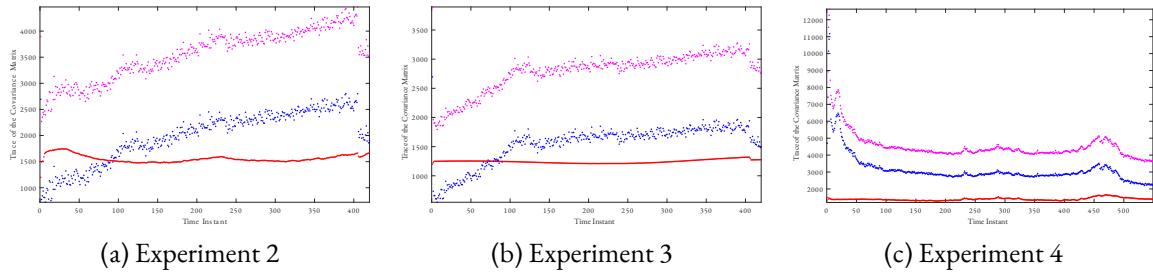


Figure 7.2: Trace of the Error-State Covariance Matrix: State variables only; Landmarks only; Whole state vector

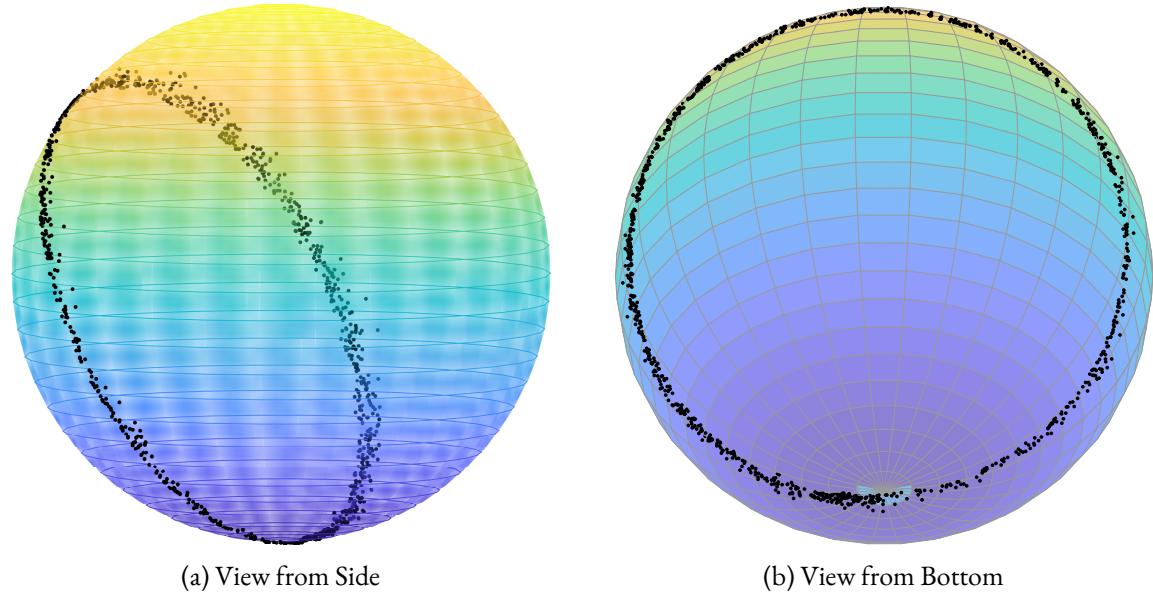


Figure 7.3: 3D Reconstruction of the Globe: reconstruction using experiment 2, all points are on the surface of the sphere, darker side in front

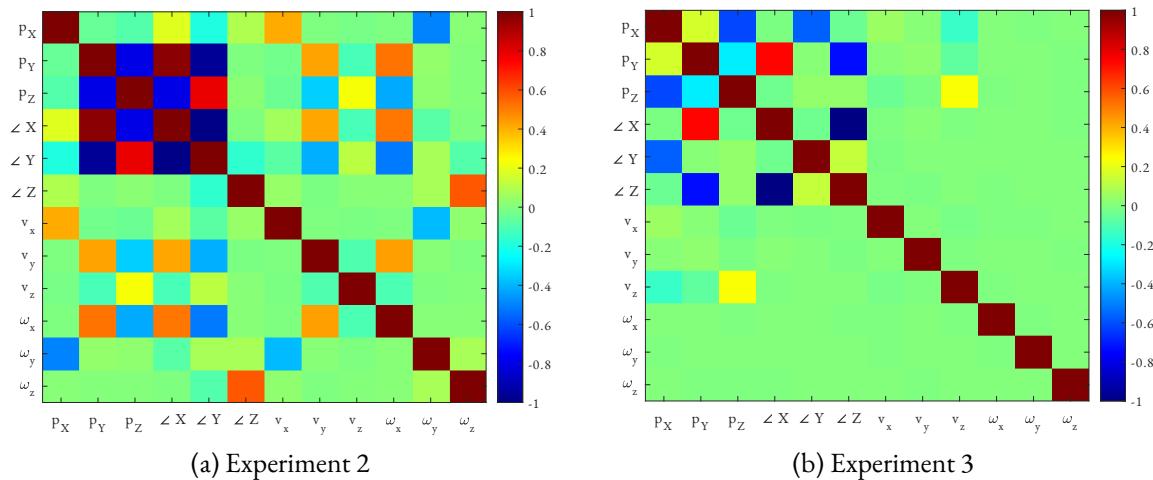


Figure 7.4: Correlation Plots: Showing difference between correlation of variable without and with presence of the gyroscope at time instant 8

Results of the fitting sphere to the reconstructed point cloud are as follows:

Table 7.2: Radius of the Fitted Sphere

Radius (mm)	Calculated (mm)	RMS Error (mm)
Experiment 2	200.028	0.0628
Experiment 3	200.009	0.0017
Experiment 4	142.749	1.9255

## 7.4 Discussions

This section discusses the results of the experiments in the context of questions asked in section 7.1, as reported in the previous section and their limitations.

### 7.4.1 Interpretation of the Results

**Is ES-EKF able to sufficiently estimate the trajectory of the camera when landmark position is already known in the world coordinate system?**

From fig. 7.1a, it is seen that the trajectory of the stereo camera system is well reconstructed using *known landmarks positions*. From the locus of azimuth, elevation and rotation angles the estimated trajectory looks like a circle of radius around 600 mm. Further, in fig. 7.1b it is also seen that velocities are constant as expected. It is also important to note that the linear velocities in the direction  $y$  and  $z$  are zero. This is due to these velocities are represented in the camera coordinate system, in which there is only lateral movement of the camera. A Similar explanation is valid for the angular velocity. Hence, experiment 1 shows that, the ES-EKF SLAM is able to correctly estimate the camera trajectory.

**Is ES-EKF SLAM able to sufficiently estimate the landmark location for 3D reconstruction and trajectory of the camera when landmark location is unknown in the world coordinate system?**

In experiment 2, the SLAM algorithm is used in conjunction with ES-EKF as an estimator. In this experiment locations of the landmarks are unknown. Hence, the additional task here is to estimate the transform  $^{ccs}T_{ccs}$ . From fig. 7.1d, it is observed that the **estimated rotation angle** leads the **reference rotation angle**; this implies, that the estimated trajectory of the camera leads the **reference trajectory**. This also means, the **estimated rotation** is too optimistic. This is due to the propagation of the initial ambiguity between the translation along  $x$ -axis and rotation along the  $y$ -axis. This ambiguity is seen in the  $V_x$  and  $\omega_y$  plots in fig. 7.1e. Further, addition of more sensor can help to reduce this ambiguity. Hence, in experiment 3 it is proposed to fuse gyroscopic information in the correction step. In figs. 7.1j and 7.1l, where experiments are performed on the real data, similar results are seen, and plots are comparable.

**Does the addition of gyroscope in simulated data show any improvement in the accuracy of reconstruction?**

From figs. 7.1g and 7.1i, it is seen that fusing angular rotation information has reduced the large initial ambiguity between  $V_x$  and  $\omega_y$ . This comparison is further aided by correlation plot of the state variables in fig. 7.4. In fig. 7.4a corresponding to experiment 2, high correlation between  $V_x$  and  $\omega_y$  can be seen, whereas, in fig. 7.4b corresponding to experiment 3 it is near zero.

Another consequence of fusing sensory information is reduced lead between reference and estimated rotational angle. In literature, this is famously known as *counteracting the drift*. Further, the addition of gyroscope decreases the initial uncertainty, which in turn reduces overall propagated uncertainty. Upon fitting the sphere to the reconstructed point cloud, it is also seen that integration of the gyroscope reduces the RMS error (refer table 7.2) and yields a better fit. Similarly, smaller trace of the covariance matrix is reported (refer fig. 7.2) in the case addition of the gyroscope.

### Can covariance matrix be used to understand the error propagation and error dynamics intuitively?

The fig. 7.2 shows the trace of the state variables, all the landmarks and full state vector (state variables + landmarks). In figs. 7.2a and 7.2b, at time instant around 410, the trace decreases suddenly due to the loop closure. Whereas, in fig. 7.2c, it occurs at time instant around 460. It is seen that the loop closure drastically reduces the uncertainties of estimation as expected.

Further, covariance matrix can be easily converted into correlation matrix and plots like fig. 7.4 obtained to understand and investigate the correlation between the state variables. Hence, using covariance matrix it is possible to intuitively understand the error dynamics.

### Is it possible to comment on the ambiguity between translation along the $x$ -axis and the rotation along the $y$ -axis by means of the covariance matrix of estimation?

Due to initial ambiguity between translation along  $x$ -axis and rotation along the  $y$ -axis the trace of state variables in fig. 7.2a is higher than the trace in fig. 7.2b. Hence, if traces of state variable is individually plotted, a large trace for both linear and rotational velocity could be attributed to this ambiguity.

Further, as already explained while addressing previous question, using correlation matrix it is possible to comment on the ambiguity between translation along the  $x$ -axis and the rotation along the  $y$ -axis.

#### 7.4.2 Limitations and Recommendations

This section highlights the limitations of the current study and sets the ground for future research. This section also provides recommendations for future research.

The main limitation of ES-EKF approach is a constant velocity assumption, which rarely holds true in the real world scenario. Further, this approach also considers a linearisation of the system around the current estimate. Thus, this method has a limitation that it cannot guarantee the globally optimal solution in the non-linear case. Hence, the other classes of SLAM algorithms viz., SLAM Based on Particle Filter (PF-SLAM), Graph-based SLAM should be studied. Further, differentiating between tracking and mapping operations by means of different threads, as done in the PTAM algorithm should also be studied. Usage of PF-SLAM will also allow modelling non-linear system and the possibility of considering higher derivatives of the state variables.

Currently, the 3D structure is computed directly from a single stereo pair rather than from adjacent frames. To compute a more accurate estimation of the trajectory, local bundle estimation (over last  $m$  frames) should be performed. As per Scaramuzza and Fraundorfer (2011), after bundle adjustment, the effects of the motion are much more alleviated. Bundle Adjustment and optimization of the pose graph is the important step in the ORB-SLAM.

By means of experiments on the simulations, the importance of the integration of gyroscope is shown. Further, the hardware should be adapted to integrate angular information from the gyroscope. It is also possible to estimate 3D motion using Sceneflow (Tarifa, 2016; Alcantarilla, Yebes, Almazán, & Bergasa, 2012b; Clipp, Jongwoo Lim, Frahm, & Pollefeys, 2010), which could be fused with existing sensory measurements to obtain more accurate and robust results.

Additionally, sophisticated landmark deletion, forking policies should also be implemented. Further, this can be done by means of application of RANSAC to reduce false matches and use descriptors of both images instead of only left image.

Using camera developed in Part I of this thesis, outputting 3D surface mesh, motion compensation by application of SLAM locally and implementing the algorithm on sophisticated hardware is also in the interest of further research.



# Chapter 8

## Conclusions

To conclude this thesis, it will be assessed to what extent the goal described in Chapter 1 is met. The overall goal was to design a stereocamera system which is able to differentiate between different induration levels of PASI, while increasing the area of reconstruction. Therefore, it was investigated with what accuracy an object with a depth variation on  $0.1\text{ mm}$  is reconstructed. In the separate experiments, a visual SLAM technique is also evaluated to increase the area of reconstructed. This goal is sub-divided into two research goals which are answered separately below.

To this extent, the first research goal was: *finding the geometrical and optical requirements for a stereo camera setup for stereo-photogrammetry to obtain the depth resolution to differentiate between different levels of PASI*. For this purpose, design parameters, both optical and geometrical, for a stereo camera setup were identified. A MATLAB App is developed to visualize the stereo-geometry of camera and other dependent parameters. On the basis of the realization of the initial prototype of the stereo camera, it is concluded that a depth resolution of at least  $0.25\text{ mm}$  can be achieved. However, due to COVID-19 restrictions system is evaluated on 3D printed workpiece rather than a real diseased site.

Following this, the next goal was: *the use of ES-EKF SLAM as a visual SLAM technique to increase the reconstruction area*. To evaluate this experiments on simulated rotation of the globe as, *simulated data*, and photographs of rotating globe as, *real data*, are performed. It can be concluded that ES-EKF SLAM with only visual information is not robust and there is an ambiguity between the translation and rotation. Further, results obtained upon the integration of gyroscope with simulated visual information show that this ambiguity is reduced. This is a strong indication of the requirement of additional sensors to counteract this ambiguity.

To conclude the overall research, a stereo imaging technique applied to obtain a 3D model of the object with the depth resolution of at least  $0.25\text{ mm}$ . Further, in separate experiments, usability of a visual SLAM algorithm with and without gyroscope integration is investigated to increase the area of reconstruction. The results obtained are an initial indication of the usability of the combined system to reconstruct large Psoriasis affected skin site with an adequate depth resolution.



# Acknowledgements

At the end of this thesis, I would like to take some time to thank all the people without whose support this assignment would never be possible. Although it is just my name on the cover, many people have contributed to this research in their particular way, and for that, I want to give them special thanks.

Firstly, to my supervisor dr.ir. Ferdi van der Heijden. I am thankful for your guidance and patience throughout the assignment. I am highly grateful for your quick and active support whenever I was stuck. I would also like to thank you for improvising on the thesis topic given difficulties faced due to COVID-19 crisis. Your insightful comments during weekly meetings surely made me work even harder.

I also want to take a moment to thank Sander Grimm for implementation of the stereo system and Stijn Wolters for the supporting work on SLAM. Further, I would like to take this opportunity to thank my friends: Akash, Aniket, Ashwin, Gaurav, Husain and Parth. You guys were by my side through this journey and I cannot thank you enough.

It is also funny how the people closest to us are mentioned in the end in such text. I want to thank my family for believing in me and being there for me not only academically but also extending their support to my life beyond masters, and finally last but not the least Aditi, for the endless supply of printouts and supporting me during all ups and downs.



# Appendix A

## Definitions

This chapter provides definitions to certain terms used throughout the literature. Terms used are categorized in three broad categories viz., Image Sensor Metrics, Lens Metrics, Stereo Geometry.

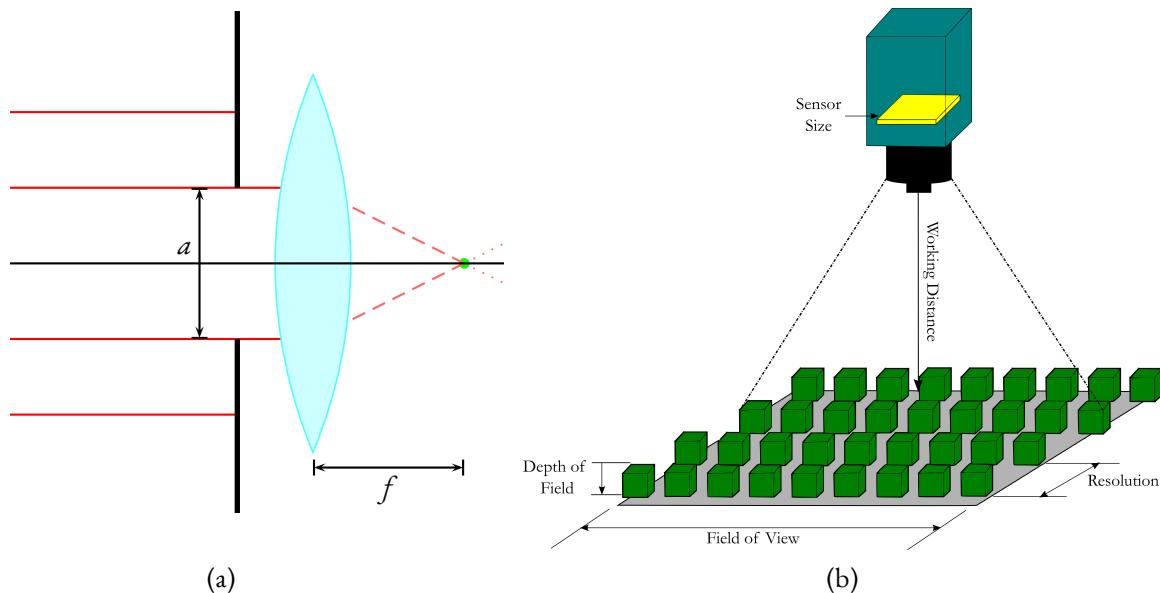


Figure A.1: **Lens and Sensor Metrics:** (a) a converging lens with aperture  $\alpha$  showing parallel rays coming from infinity meet at focal point  $f$ ; (b) bird's eye view of various sensor metrics, adapted from ([National Instruments, 2020](#)).

### A.1 Image Sensor Metrics

**Sensor Resolution** number of pixels in the image, expressed here in  $px$ .

**Sensor Size** the physical size of the sensor, depends on the pixel size, expressed here in  $mm$ .

**Sensor Format** the physical size of the sensor, but is not dependent on the pixel size, often expressed in inches.

**Pixel Size** size of the individual pixel, generally expressed in  $\mu m$ .

## A.2 Lens Metrics

**Focal Length** rays of light entering a converging lens parallel to its axis converge at its focal point F, distance between optical centre and F is known as focal length, denoted by  $f$  and expressed in  $\text{mm}$ . Sometimes, focal length is also expressed in pixels, in that case it is denoted as  $f_{\text{px}}$ . It is important to note that, the focal length mentioned on camera is true only when camera is focused at infinity.

**Aperture** the opening in a lens through which light passes to enter the camera, expressed here in  $\text{mm}$  or in *inches* in case of f-number of f-stop.

**Angle of View** describes the angular extent of a given scene that is imaged by a camera, expressed in degrees.

**Field of View** area under inspection that the camera needs to acquire, expressed here in  $\text{mm}^2$ .

**Hyperfocal Distance** distance beyond which all objects can be brought into an *acceptable* focus

**Near and Far limit** near point is the closest point at which an object can be placed and still form a focused image on the retina and far point is farthest point at which an object can be placed and still form a focused image on the retina. Expressed in  $\text{mm}$

## A.3 Stereo Geometry

**Baseline Distance** distance between principal points of two cameras, expressed in  $\text{mm}$ .

**Working Distance** distance from the front of the lens to the object under inspection, expressed here in  $\text{mm}$ .

**Depth Resolution** the minimum distance possible between two adjacent depth values, expressed here  $\text{mm}$ .

## Appendix B

### PASI and PASImeter

Psoriasis Area and Severity Index represents a summary or composite score of a variety of physical signs of psoriasis and is currently the most often cited instrument of this type. PASI is calculated using a complex formula for rating the four body regions, viz., the head, trunk, upper extremities, and lower extremities. For each region, the lesion area, the degree of erythema (redness), induration (thickness), and desquamation (scaliness) of the plaques are determined. The scores from the regions are weighted and summed to give a PASI score ranging from 0 – 72 as follows ([Ashcroft, Po, Williams, & Griffiths, 1999](#); [Hurriyatul et al., 2010](#)):

$$\begin{aligned} \text{PASI} = & 0.1 \times (R_h + T_h + S_h) \times A_h + \\ & 0.2 \times (R_u + T_u + S_u) \times A_u + \\ & 0.3 \times (R_t + T_t + S_t) \times A_t + \\ & 0.4 \times (R_l + T_l + S_l) \times A_l \end{aligned}$$

Table B.1: **PASI Explanation**

Erythema (R)			Surface Area (A)			Subscripts	
Score	Grade	Score	Grade	Letter	Body Region	Weightage	
0	None	0	None	h	Head	0.1	
1	Mild	1	1% to 9%	u	Upper extremities	0.2	
2	Moderate	2	10% to 29%	t	Trunk	0.3	
3	Severe	3	30% to 49%	l	Lower extremities	0.4	
4	Very Severe	4	50% to 69%				
		5	70% to 89%				
		6	90% to 100%				



## Appendix C

# Workpiece Drawing

This appendix provides the CAD diagrams of the designed workpiece. Initially isometric view is presented followed by a technical diagram.

This workpiece is designed using FreeCAD<sup>1</sup>.

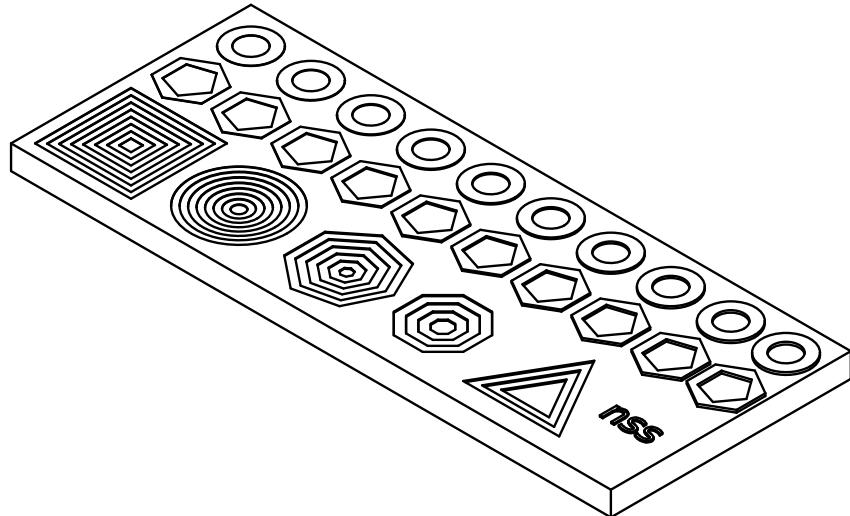
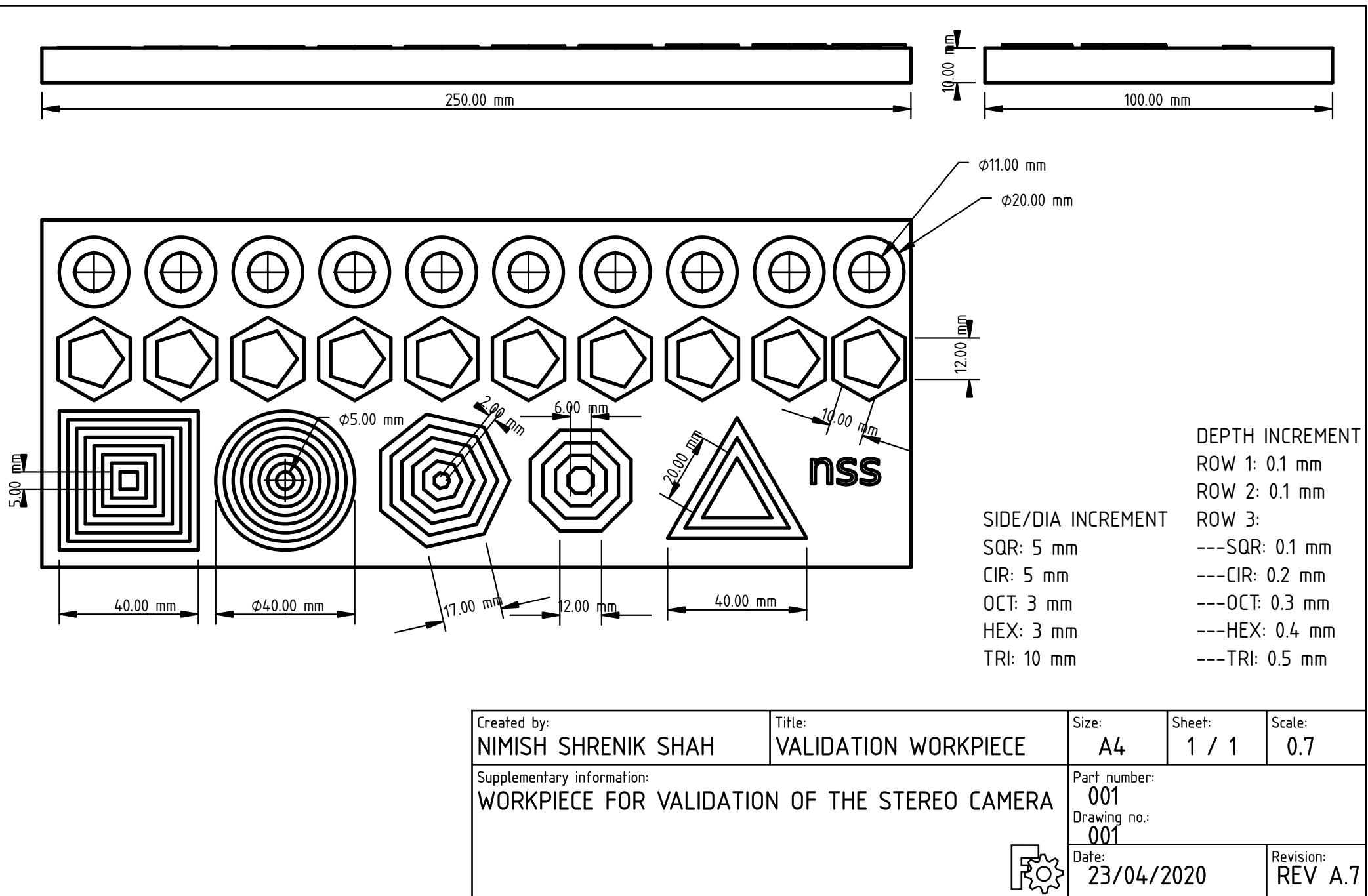


Figure C.1: Isometric View of the Validation Workpiece

---

<sup>1</sup><https://www.freecadweb.org/>



# References

- Ahmad Fadzil, M. H., Prakasa, E., Asirvadam, V. S., Nugroho, H., Affandi, A. M., & Hussein, S. H. (2013). 3d surface roughness measurement for scaliness scoring of psoriasis lesions. *Computers in Biology and Medicine*, 43(11), 1987 - 2000. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0010482513002242> doi: <https://doi.org/10.1016/j.compbiomed.2013.08.009>
- Alcantarilla, P. F., Yebes, J. J., Almazán, J., & Bergasa, L. M. (2012a, May). On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 ieee international conference on robotics and automation* (p. 1290-1297). doi: 10.1109/ICRA.2012.6224690
- Alcantarilla, P. F., Yebes, J. J., Almazán, J., & Bergasa, L. M. (2012b, May). On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 ieee international conference on robotics and automation* (p. 1290-1297). doi: 10.1109/ICRA.2012.6224690
- Ashcroft, Po, L. W., Williams, & Griffiths. (1999). Clinical measures of disease severity and outcome in psoriasis: a critical appraisal of their quality. *British Journal of Dermatology*, 141(2), 185-191. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2133.1999.02963.x> doi: 10.1046/j.1365-2133.1999.02963.x
- Bang, J., Lee, D., Kim, Y., & Lee, H. (2017, Feb). Camera pose estimation using optical flow and orb descriptor in slam-based mobile ar game. In *2017 international conference on platform technology and service (platcon)* (p. 1-4). doi: 10.1109/PlatCon.2017.7883693
- Bar-Shalom, Y., & Li, X.-R. (1993). Estimation and tracking- principles, techniques, and software. *Norwood, MA: Artech House, Inc, 1993..*
- Basra, M. K., & Shahrukh, M. (2009). Burden of skin diseases. *Expert Review of Pharmacoeconomics & Outcomes Research*, 9(3), 271-283. Retrieved from <https://doi.org/10.1586/erp.09.23> doi: 10.1586/erp.09.23
- Bloemen, M. C., van Gerven, M. S., van der Wal, M. B., Verhaegen, P. D., & Middelkoop, E. (2011). An objective device for measuring surface roughness of skin and scars. *Journal of the American Academy of Dermatology*, 64(4), 706 - 715. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0190962210003063> doi: <https://doi.org/10.1016/j.jaad.2010.03.006>
- cfr. (2016). *More elegant way to achieve this same camera perspective projection model?* Retrieved April 1, 2020, from <https://tex.stackexchange.com/a/323778>
- Cheng, J., Sun, Y., & Meng, M. Q.-H. (2019). Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 33(12), 576-589. Retrieved from <https://doi.org/10.1080/01691864.2019.1610060> doi: 10.1080/01691864.2019.1610060

Clipp, B., Jongwoo Lim, Frahm, J., & Pollefeys, M. (2010, Oct). Parallel, real-time visual slam. In *2010 ieee/rsj international conference on intelligent robots and systems* (p. 3961-3968). doi: 10.1109/IROS.2010.5653696

Corke, P. (2017). *Robotics, vision and control: fundamental algorithms in MATLAB® second, completely revised* (Vol. 118). Springer.

Fadzil, M. A., Prakasa, E., Fitriyah, H., Nugroho, H., Affandi, A. M., & Hussein, S. H. (2010). Validation on 3d surface roughness algorithm for measuring roughness of psoriasis lesion. *Biological and Biomedical Sciences*, 7(4), 205–2010.

Fang, W., Zheng, L., & Deng, H. (2016, Nov). A motion tracking method by combining the imu and camera in mobile devices. In *2016 10th international conference on sensing technology (icst)* (p. 1-6). doi: 10.1109/ICSENS-T.2016.7796235

Fleischer, A. B., Feldman, S. R., Rapp, S. R., Reboussin, D. M., Exum, M. L., Clark, A. R., & Rajashekhar, V. (1996). Disease severity measures in a population of psoriasis patients: The symptoms of psoriasis correlate with self-administered psoriasis area severity index scores. *Journal of Investigative Dermatology*, 107(1), 26 - 29. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0022202X15426193> doi: <https://doi.org/10.1111/1523-1747.ep12297659>

Gao, J., Yang, Y., Lin, P., & Park, D. S. (2018). Computer Vision in Healthcare Applications. *Journal of healthcare engineering*, 2018.

Grasa, O. G., Civera, J., Guemes, A., Munoz, V., & Montiel, J. (2009). Ekf monocular slam 3d modeling, measuring and augmented reality from endoscope image sequences. In *Medical image computing and computer-assisted intervention (miccai)* (Vol. 2). Retrieved from [http://webdiis.unizar.es/~jcivera/papers/garcia\\_etal\\_miccai09.pdf](http://webdiis.unizar.es/~jcivera/papers/garcia_etal_miccai09.pdf)

Grimm, S. (2020). *Design of a 3d imaging system for psoriasis assesment* (Master's thesis). Retrieved from <http://essay.utwente.nl/83959/>

Hani, A. F. M., & Prakasa, E. (2014, June). *Methodology and apparatus for objective, non-invasive and in vivo assessment and rating of psoriasis lesion scaliness using digital imaging*. Google Patents. (US Patent 8,755,577)

Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (Second ed.). Cambridge University Press, ISBN: 0521540518.

Hay, R. J., Johns, N. E., Williams, H. C., Bolliger, I. W., Dellavalle, R. P., Margolis, D. J., ... Naghavi, M. (2014). The global burden of skin disease in 2010: An analysis of the prevalence and impact of skin conditions. *Journal of Investigative Dermatology*, 134(6), 1527 - 1534. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0022202X15368275> doi: <https://doi.org/10.1038/jid.2013.446>

Howard, I. P. (2012). *Perceiving in depth, volume 1: Basic mechanisms*. Oxford University Press.

Hurriyatul, F., Fadzil, A., & Hani, M. (2010). Objective assessment of psoriasis lesion thickness for pasi scoring 3d digital imaging. *Proceedings of World Academy of Science, Engineering and Technology* 63 2010, 63, 109–115.

Jacobi, U., Chen, M., Frankowski, G., Sinkgraven, R., Hund, M., Rzany, B., ... Lademann, J. (2004). In vivo determination of skin surface topography using an optical 3d device. *Skin Research and Technology*, 10(4), 207-214. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1600-0846.2004.00075.x> doi: 10.1111/j.1600-0846.2004.00075.x

Langley, R. G., & Ellis, C. N. (2004). Evaluating psoriasis with psoriasis area and severity index, psoriasis global assessment, and lattice system physician's global assessment. *Journal of the American Academy of Dermatology*, 51(4), 563 - 569. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0190962204010886> doi: <https://doi.org/10.1016/j.jaad.2004.04.012>

Lee, O., Lee, K., Oh, C., Kim, K., & Kim, M. (2014). Prototype tactile feedback system for examination by skin touch. *Skin Research and Technology*, 20(3), 307-314. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1111/srt.12120> doi: 10.1111/srt.12120

Madhyastha, V., Ravindra, V., Mallikarjunan, S., & Goyal, A. (2011). Extended kalman filter vs. error state kalman filter for aircraft attitude estimation. In *Aiaa guidance, navigation, and control conference* (p. 6615). Retrieved from <https://arc.aiaa.org/doi/abs/10.2514/6.2011-6615> doi: 10.2514/6.2011-6615

Mahmoud, N., Grasa, Ó. G., Nicolau, S. A., Doignon, C., Soler, L., Marescaux, J., & Montiel, J. M. M. (2017, Jan 01). On-patient see-through augmented reality based on visual slam. *International Journal of Computer Assisted Radiology and Surgery*, 12(1), 1-11. Retrieved from <https://doi.org/10.1007/s11548-016-1444-x> doi: 10.1007/s11548-016-1444-x

Mourikis, A. I., & Roumeliotis, S. I. (2007, April). A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 ieee international conference on robotics and automation* (p. 3565-3572). doi: 10.1109/ROBOT.2007.364024

Munro, P., & Gerdelen, A. P. (2009). Stereo vision computer depth perception. *Country United States City University Park Country Code US Post code*, 16802.

National Instruments. (2020). *Calculating camera sensor resolution and lens focal length*. Retrieved April 7, 2020, from <https://www.ni.com/nl-nl/support/documentation/supplemental/18/calculating-camera-sensor-resolution-and-lens-focal-length.html>

Nister, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 756-770.

Ogata, K. (1995). *Discrete-time control systems*. Englewood Cliffs, NJ: Prentice Hall.

Ogata, K. (2010). *Modern control engineering*. Boston: Prentice-Hall.

Park, S. Y., Ha, S.-H., Yu, D.-S., Son, S.-W., Kim, D.-J., Kim, I.-H., ... Oh, C.-H. (2004). Quantitative evaluation of severity in psoriatic lesions using three-dimensional morphometry. *Experimental Dermatology*, 13(4), 223-228. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0906-6705.2004.00122.x> doi: 10.1111/j.0906-6705.2004.00122.x

Parwana, H., & Kothari, M. (2017). *Quaternions and attitude representation*. Retrieved from <https://arxiv.org/abs/1708.08680>

Prakash, D. (2007). *Stereoscopic 3D viewing systems using a single sensor camera* (Master's thesis, The Ohio State University). Retrieved from [http://rave.ohiolink.edu/etdc/view?acc\\_num=osu1196268883](http://rave.ohiolink.edu/etdc/view?acc_num=osu1196268883)

Prof. Dr. med. Kristian Reich, P. D. m. M. A. (2013). Psoriasis area severity index meter (pasimeter) [Computer software manual]. Max-Planck-Ring 2a, 65205 Wiesbaden, Germany. Retrieved from <https://www.dermatools.net/dam/jcr:8cbb4e58-dcd3-4144-b6ae-b6f209dea3f8/PASIMETER.pdf>

Riisgaard, S., & Blas, M. R. (2003). *Slam for dummies* (Tech. Rep.). MIT OpenCourseWare.

Scaramuzza, D., & Fraundorfer, F. (2011, Dec). Visual odometry [tutorial]. *IEEE Robotics Automation Magazine*, 18(4), 80-92. doi: 10.1109/MRA.2011.943233

Solà, J. (2014, October). *Simultaneous localization and mapping with the extended kalman filter* (Tech. Rep.). Retrieved from [https://jinyongjeong.github.io/images/post/SLAM/lec05\\_EKF\\_SLAM/EKF.pdf](https://jinyongjeong.github.io/images/post/SLAM/lec05_EKF_SLAM/EKF.pdf)

Solà, J. (2017). Quaternion kinematics for the error-state kalman filter. *CoRR, abs/1711.02508*. Retrieved from <http://arxiv.org/abs/1711.02508>

Solin, A., Cortes, S., Rahtu, E., & Kannala, J. (2018, July). Inertial odometry on handheld smartphones. In *2018 21st international conference on information fusion (fusion)* (p. 1-5). doi: 10.23919/ICIF.2018.8455482

Sturm, P., Ramalingam, S., Tardif, J.-P., Gasparini, S., & Barreto, J. (2011). Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(1–2), 1-183. Retrieved from <http://dx.doi.org/10.1561/0600000023> doi: 10.1561/0600000023

Taketomi, T., Uchiyama, H., & Ikeda, S. (2017, Jun 02). Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1), 16. Retrieved from <https://doi.org/10.1186/s41074-017-0027-2> doi: 10.1186/s41074-017-0027-2

Tarifa, M. J. (2016). *Scene flow*. MAchine Perception and Intelligent Robotics (MAPIR), Universidad de Málaga. Retrieved from <http://mapir.isa.uma.es/mapirwebsite/index.php/computer-vision-menu-topic/207-scene-flow-topic.html>

The National Institute for Rehabilitation Engineering. (2004). *Visual distance perception & depth perception*. Retrieved April 3, 2020, from <https://www.schepens.harvard.edu/images/stories/nire/distance-perception.pdf>

Tola, E., Lepetit, V., & Fua, P. (2008, June). A fast local descriptor for dense matching. In *2008 ieee conference on computer vision and pattern recognition* (p. 1-8). doi: 10.1109/CVPR.2008.4587673

Tola, E., Lepetit, V., & Fua, P. (2010, May). Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5), 815-830. doi: 10.1109/TPAMI.2009.77

van der Heijden, F. (2019a). *3D measurements from camera images* (Tech. Rep.). Control Laboratory, EL/RAM, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands: Robotics and Mechatronics, University of Twente. (Syllabus for course 3D Computer Vision for Medical Applications)

van der Heijden, F. (2019b). *Camera models* (Tech. Rep.). Control Laboratory, EL/RAM, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands: Robotics and Mechatronics, University of Twente. (Syllabus for course 3D Computer Vision for Medical Applications)

van der Heijden, F. (2019c). *Math for computer vision and navigation* (Tech. Rep.). Control Laboratory, EL/RAM, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands: Robotics and Mechatronics, University of Twente. (Syllabus for course 3D Computer Vision for Medical Applications)

van der Heijden, F. (2019d). *Stereo: from pixels to 3D surface mesh* (Tech. Rep.). Control Laboratory, EL/RAM, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands: Robotics and Mechatronics, University of Twente. (Syllabus for course 3D Computer Vision for Medical Applications)

van der Heijden, F. (2020). *Visual Navigation* (Tech. Rep.). Control Laboratory, EL/RAM, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands: Robotics and Mechatronics, University of Twente. (Syllabus for course 3D Computer Vision for Medical Applications)

Van Der Heijden, F., Duin, R. P., De Ridder, D., & Tax, D. M. (2005a). Parameter estimation. In *Classification, parameter estimation and state estimation* (p. 45-80). John Wiley & Sons, Ltd. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470090154.ch3> doi: 10.1002/0470090154.ch3

Van Der Heijden, F., Duin, R. P., De Ridder, D., & Tax, D. M. (2005b). State estimation in practice. In *Classification, parameter estimation and state estimation* (p. 253-308). John Wiley & Sons, Ltd. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/0470090154.ch8> doi: 10.1002/0470090154.ch8

van der Valk, D. (2019). *Rpi-based passive 5-camera system for 3D face acquisition* (Master's thesis, University of Twente). Retrieved from <http://essay.utwente.nl/79026/>

Vivid Vision. (n.d.). *Depth perception*. Retrieved April 3, 2020, from [https://www.seevidly.com/info/Binocular\\_Vision/Visual\\_Skills/Depth\\_Perception](https://www.seevidly.com/info/Binocular_Vision/Visual_Skills/Depth_Perception)

Wikipedia contributors. (2020). *Angle of view — Wikipedia, the free encyclopedia*. Retrieved April 7, 2020, from [https://en.wikipedia.org/w/index.php?title=Angle\\_of\\_view&oldid=977037523](https://en.wikipedia.org/w/index.php?title=Angle_of_view&oldid=977037523) (permalink)

Williams, S., & Wilkins, J. (2005). *Pathophysiology: 2 in 1 reference for the nurses*. Philadelphia: Lippincott.

World Health Organization. (2016). *Global report on psoriasis* [Publications]. World Health Organization. Retrieved from <https://apps.who.int/iris/handle/10665/204417>