



Norwegian University of
Science and Technology

Study of wind turbine wake aerodynamics through the application of motion tracking techniques

Jenny Marie Bjørnsgaard

Master of Energy and Environmental Engineering

Submission date: June 2016

Supervisor: Nicholas Worth, EPT

Norwegian University of Science and Technology
Department of Energy and Process Engineering

EPT-M-2016-15

MASTER THESIS

for

Student Jenny Marie Bjørnsgaard

Spring 2016

Study of wind turbine wake aerodynamics through the application of motion tracking techniques**Background and objective**

The use of wind energy is becoming increasingly widespread, and is likely to feature heavily in sustainable energy networks of the future. However, the complex aerodynamic performance of wind turbines result in significant challenges during their design and deployment. The existing extensive use of two-dimensional design methods (such as as the XFOIL code) to design systems which can under certain operating conditions contain strong three-dimensional effects may lead to poor (or even unknown) performance at those conditions, altering the fatigue life and power generated. This project will focus on setting up a novel experiment to investigate these effects based on rotating and translating a simplified model wing through a tank of water, and visualising the flow over the wing.

The project will initially focus on the numerical design of an optical tracking system in order to evaluate the 3D motion of the wing through the water. This will involve the use of perspective projection models and methods in order to track points of interest. The project will also involve a simple experimental test of this approach and accuracy assessment. Tracking measurements will be made in the new experimental facility, and an assessment of the wing translation path will be made, with particular attention to deviations from the proscribed trajectory. Time permitting; flow visualisation work will also be conducted to visualize wing tip vortices and the dependence of these on the time-varying trajectory of the wings.

The following tasks are to be considered:*Literature review*

1. Background reading and literature review of wind turbine wakes, including: why studying these is of interest in a practical sense; the dominant structure of the wake and coherent motions within it; the formation of tip vortices; the role of tip vortices in modifying the turbulent structure of the wake
2. Background reading and literature review of imaging, camera calibration models, particle (or point) tracking in 3D

3D tracking activities (preparation for translating wing experiments)

1. Investigate camera calibration with Matlab computer vision toolbox, including bundle adjustment for multiple cameras
2. Investigate (using either numerical or some real data with multiple cameras) 3D point location and tracking
3. Set up a simple tracking experiment using a linear traverse and a marked object, and track in 3D using multiple cameras. Assess the accuracy of the tracking and report

Translating wing experiment

1. Track the wing in the tank for a number of cases, to investigate trajectory, stability and frequency of oscillations
2. Setup a flow visualisation experiment, whereby the flow structure will be visualised using dye fed to strategic locations on the wing.
3. Relate flow structures to wing structure deflections to assess the impact of a flexible structure on the formation of vortices.

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report. If the documentation on risk assessment represents a large number of pages, the full version is to be submitted electronically to the supervisor and an excerpt is included in the report.

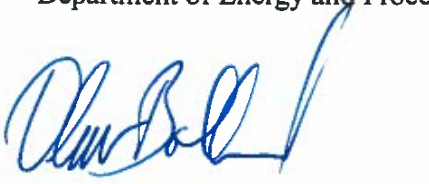
Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

Work to be done in lab (Water power lab, Fluids engineering lab, Thermal engineering lab)

Field work

Department of Energy and Process Engineering, 14. January 2016



Olav Bolland
Department Head



Nicholas Worth
Academic Supervisor

Research Advisor:

Preface

The work in this master thesis was performed at the Norwegian University of Science and Technology (NTNU) during spring of 2016 for the Department of Energy and Process Engineering. The experimental work was carried out in the Aerodynamic laboratory at NTNU.

The main purpose of the master thesis was to reconstruct the trajectories of a rotating object in space with cameras, and estimate the accuracy of the reconstruction. The study will be applied in a wind turbine experimental facility at NTNU, that was omitted in this theses due to construction delay.

I would like to thank my supervisor Nicholas Worth, for being patient with me and giving me great help during many frustrating hours. He guided me in the right direction and had always new and interesting ideas for my work.

I would also like to thank the staff at the Aerodynamic laboratory and Thermal Energy Building for providing me with equipment and assistance. Special thanks to Arnt Egil Kolstad who was always helpful and positive when I needed materials constructed for my laboratory work.

Last but not least, I would like to thank Marcus, my friends and my family. They supported me and helped me relax when my mind needed time out. My master would have been difficult to accomplish without you.

Trondheim, 10-06-2016



Jenny Marie Bjørnsgaard

Executive summary

In this master thesis an optical tracking system used to evaluate the 3D motion of an object in space has been studied. The aim of the study was to reconstruct the path of a rotating plank, and evaluate the accuracy of the reconstruction. The tracking system involved the use of cameras making perspective modelling and camera calibration necessary. Accuracy of reconstructing world positions from 2D image coordinates has been established through both numerical and experimental investigation. A simple tracking experiment was performed on a rotating plank, to determine deviations from its proscribed trajectory. The tracking method was to be applied in an experimental facility in the Aerodynamic laboratory at NTNU, but was omitted due to construction delay during the period of this thesis work.

Cameras were calibrated using functions in MATLAB Computer Vision Toolbox. Reconstruction errors were determined through back projection of calibration points. The calibration parameters were thereafter used in a numerical analysis to estimate optimal camera positions about the frame object. Monte Carlo simulations were performed with varying angles between setups with two, three and four cameras, respectively. Last, a point tracking experiment was conducted by rotating a plank marked with a traceable point. Two cameras on a linear traverse were used, varying the angle between them and the angle of the object position.

Mean reconstruction errors from the calibration experiment resulted in 0.05 mm in x -direction, 0.04 mm in y -direction, and 0.1 mm in z -direction. From the numerical analysis, the angle of 45° in vertical direction between each camera and the horizontal plane resulted to be the optimal positioning for all combinations. This implies a total angle of 90° between two cameras. Furthermore, the setup with four cameras resulted in most accurate reconstruction with a maximum error of only 0.13 mm. In the point tracking experiments, the radii of the rotating paths were estimated within 99.4% of actual radius for non angled plank; 99.1% for 20° and 35° angled plank; and 98.4% for 45° angled plank. Fluctuations of depth was registered in all tracking sequences. Vibration of plank was visually registered, but the magnitude of the vibration was difficult to estimate.

Sammendrag

I denne masteroppgaven har et optisk sporingssystem for å kartlegge det tredimensjonale bevegelsesmønsteret til et roterende objekt blitt studert. Sporingssystemet innebærer bruken av kameraer som viktigjør både modellering, samt kalibrering av kameraer. Nøyaktighet av å rekonstruere posisjoner i rommet fra todimensjonale bildekoordinater har derfor blitt gjennomført både ved numeriske analyser og ved eksperimentelle forsøk. Et enkelt sporingseksperiment er utført ved å rotere en planke markert med et punkt, for å bestemme avvik fra dens opprinnelige plane bane. Sporingsmetoden skulle opprinnelig bli anvendt i en eksperimentell rigg ved Institutt for Energi og Prosessteknikk, men måtte utelates i denne masteroppgaven grunnet forsinkelser ved oppbygging av riggen.

Kalibrering av kameraer ble gjennomført av funksjoner i MATLAB Computer Vision Toolbox. Rekonstrueringsfeil ble bestemt via tilbakeprojisering av kalibreringspunkter. Kameraparameterene ble deretter brukt i numerisk analyse for å estimere optimale kameraposisjoner om objektet. Monte Carlo simuleringer ble gjennomført med variende vinkler mellom oppsettene av henholdsvis to, tre og fire kameraer. Til slutt ble et sporingseksperiment gjennomført ved å rotere en planke markert med et sporbart punkt. To kameraer plassert på en horisontal travers ble brukt. Både vinkelen mellom kameraene og vinkelen på planken varierte.

Gjennomsnittlige rekonstrueringsfeil fra kalibreringseksperimentet resulterte i henholdsvis 0.05 mm i x -retning, 0.04 mm i y -retning, og 0.1 mm i z -retning. Fra den numeriske analysen resulterte optimal vinkel mellom kamera og horisontalt plan 45° for alle testede kombinasjoner. Dette betyr en total vinkel på 90° mellom to kameraer. I tillegg viste oppsettet med fire kameraer seg å resultere i best nøyaktighet med kun 0.13 mm som høyeste rekonstruksjonsfeil i dybderetning. I punktsporingseksperimentet ble radiusene av de rekonstruerte banene estimert innenfor 99.4% av faktisk baneradius der planken ikke var vinklet, mens estimert til 99.1% for 20° og 35° vinklet planke og 98.4% for 45° vinkling. Svingninger rundt dybdeestimering ble registrert i alle sporingsekvensene. Vibrasjon av planken i dybderetning ble registrert visuelt, men størrelsen på vibrasjonen var vanskelig å klassifisere.

Contents

Preface	i
Summary and Conclusions	iii
Sammendrag	iv
List of figures	xi
List of tables	xiii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem formulation	3
1.2.1 Previous work	4
1.3 Thesis outline	5
2 Theory	7
2.1 Camera modelling	8
2.1.1 Pinhole camera model and perspective projection	8
2.1.2 Intrinsic and extrinsic camera parameters	10
2.1.3 Lens distortion	13
2.2 Camera Calibration	15
2.2.1 2D calibration	16
2.2.2 3D calibration	16
2.3 Reconstruct world position	18
2.4 Point Tracking	21
2.5 Apparatus and Electronics	23

3 Methodology	25
3.1 Camera calibration	25
3.1.1 Experimental camera calibration	26
3.2 Numerical accuracy analysis	28
3.3 Point Tracking	32
3.3.1 Experimental point tracking	33
4 Results and Discussion	37
4.1 Experimental camera calibration	37
4.1.1 Discussion camera calibration	41
4.2 Numerical angle analysis	41
4.2.1 Discussion numerical analysis	46
4.3 Point tracking experiment	47
4.3.1 Arc of rotation	47
4.3.2 Discussion of arc tracking	52
4.3.3 Full rotation tracking	53
4.3.4 Discussion of full rotation tracking	56
4.4 Uncertainty analysis	57
5 Conclusion	59
5.1 Conclusions	59
5.2 Recommendations for Further Work	61
5.2.1 Water tank experiment	61
5.2.2 Tracking enhancement	64
Bibliography	66
A Additional Information	i
A.1 Derivation of equation 2.15 in section 2.3	i
A.2 Setting parameters of <code>imfindcircles</code>	ii
B Numerical parameters	iii

C	MATLAB and Labview scripts	ix
C.1	MATLAB camera calibration code	ix
C.2	MATLAB numerical analysis code	xi
C.3	MATLAB point tracking code	xiv
C.4	Labview script step motor	xvi
D	Risk assessment	xix

List of Figures

1.1	Global cumulative installed wind capacity from 1997 to 2014 [24]	2
1.2	Wake structures [1]	3
2.1	Illustrations of additional camera models	8
2.2	Pinhole camera model	9
2.3	Change of coordinate system in the retinal plane	10
2.4	Change of world reference frame	12
2.5	Example of lens distortion of object with the distorted image to the left and undistorted image to the right [8]	14
2.6	Examples of 2.6a 2D and 2.6b 3D calibration objects	15
2.7	Flow chart of 2D camera calibration	17
2.8	Flow chart of 3D calibration steps	18
2.9	Illustration of triangulation with two cameras: the same point, x is identified in frames from two cameras as P1 and P2 .	19
2.10	Illustration of closest intersection point of almost intersecting lines of sight, from Mann and Ott [13]	20
2.11	Camera and lens used in experiments	23
3.1	Object fastened to scaffold used for calibration	26
3.2	Experimental setup for the camera calibration	27
3.3	Illustration of spherical coordinates together cartesian coordinates [28]	29
3.4	Camera position of two cameras	30
3.5	Asymmetric camera position of three cameras	31

3.6	Symmetric camera position of three cameras	31
3.7	Asymmetric camera position of four cameras	31
3.8	Symmetric camera position of four cameras	32
3.9	Sketch of plank used for tracking experiment. Dimensions are given on sketch. . .	33
3.10	Setups of tracking experiments	34
3.11	Too many circles(a) and correct circle(b) detected in image from tracking sequence	35
4.1	Approach (A). Histograms of errors in x, y and z-direction obtained from capturing one object orientation from all camera positions at a time	38
4.2	Approach (B). Histograms of errors in x, y and z-direction from the procedure where all object orientations are captured from one camera position at a time . . .	39
4.3	Contour plots of total reconstruction errors from images used in the camera calibration	40
4.4	Mean error contour for corresponding angles α and θ for a two-camera setup . . .	42
4.5	Mean error contour for corresponding angles α and θ for an asymmetric three-camera setup, see figure 3.5	43
4.6	Mean error contour for corresponding angles α and θ for a symmetric three-camera setup, see figure 3.6	44
4.7	Mean error contour for corresponding angles α and θ for an asymmetric four-camera setup, see figure 3.7	44
4.8	Mean error contour for corresponding angles α and θ for a symmetric four-camera setup, see figure 3.8	45
4.9	Total rms error of a 2 camera setup	45
4.10	Total rms error for 3 cameras in; (a) asymmetric setup; and (b) symmetric setup for corresponding α and θ angle.	46
4.11	Total rms error for 4 cameras in; (a) asymmetric setup; and (b) symmetric setup for corresponding α and θ angles	46
4.12	Projected paths of sequences of Arc 1	49
4.13	Fluctuating depth values of reconstructed points in sequences of Arc 1	49
4.14	Projected paths of sequences of Arc 2	50

4.15 Fluctuating depth values of reconstructed points in sequences of Arc 2	51
4.16 Projected paths of sequences of Arc 3	51
4.17 Fluctuating depth values of reconstructed points in sequences of Arc 3	52
4.18 x and y coordinates of rotated tracking results in xy -plane.	54
4.19 Fluctuating depth reconstruction of full rotation tracking	55
4.20 Illusration of skewed tracking point when viewed from angle	56
5.1 Experimental Towing tank rig	62

List of Tables

3.1	Position and orientation of camera on traverse	27
3.2	Orientation of calibration object	28
3.3	Input values for numerical angle study for camera calibration accuracy	29
3.4	Different positions of cameras and plank for point tracking experiment	34
4.1	Table of camera calibration results	37
4.2	Minimum and maximum mean error values	43
4.3	Tracking results from reconstruction of arc	48
4.4	Tracking results from reconstruction of circular path	53
B.1	Rotation matrices and camera coordinates for numerical angle study of two cameras	iv
B.2	Rotation matrices and camera coordinates for numerical angle study of three cameras located in a symmetric triangle	v
B.3	Rotation matrices and camera coordinates for numerical angle study of three cameras located in an asymmetric triangle	vi
B.4	Rotation matrices and camera coordinates for numerical angle study of four cameras located symmetrically	vii
B.5	Rotation matrices and camera coordinates for numerical angle study of four cameras located asymmetrically	viii

Nomenclature

Acronyms

ABL Atmospheric boundary layer

CCD Charge-coupled device

CID Charge injection device

CMOS Complimentary metal-oxide-semiconductors - active pixel sensors

PFV Photron FASTCAM Viewer

PIV Particle Image Velocimetry

PTV Particle Tracking Velocimetry

fps Frames per second

Greek

α Size of focal length in horizontal pixels; Rotation of yz-plane

β Size of focal length in vertical pixels

γ Skew factor

λ Scaling factor

θ Rotation of xz-plane

$\theta_i, \theta_j, \theta_k$ Euler angles

Roman

A Intrinsic camera matrix

C, c Optical camera center/principle point

H Homography of projection matrix of size 3x3

m Image point in projective aspect

R Rotation matrix

t Translation vector

$\tilde{\mathbf{m}}$ Homogeneous image coordinates

$\tilde{\mathbf{P}}$ Projective projection matrix

\tilde{M} Homogeneous world coordinates

f Focal length

M World point in projective aspect

u, v image coordinates

X, Y, Z world coordinates

Other

\mathcal{F} Focal Plane

\mathcal{P}^n Projective plane ($n=2$) or space ($n=3$)

\mathcal{R} Retinal Plane

Chapter 1

Introduction

1.1 Background and Motivation

Using the wind as an energy resource have been performed by humans for centuries. Exploiting the wind energy in transportation of sailing boats, pumping water and grinding grain with windmills, are all examples from before the electric power production became of interest.

The first windmills used for electricity production came in the late 19th century in Scotland, Denmark and United States. However, the invention of technology to convert fossil fuels to useful energy made the role of the wind as an energy resource insignificant until the mid 20th century, when the knowledge of limited fossil fuel resources became apparent in the late 1960s. The wind energy industry was then brought back on the table, and by the 1990s it saw a strong resurgence as the total worldwide installed capacity increased significantly [14].

The increasing energy demand and focus on energy production by renewable energy sources, have caused the interest in wind energy to grow rapidly during the past decades. The global status of 2014 resulted in a total capacity of approximately 369.6 GW. For comparison, the total capacity in 2013 was 318.1 GW, making the year of 2014 a record year with over 51 GW total installed capacity around the world [24]. Considering the cumulative installed capacity from earlier years depicted in figure 1.1, the installation will continue to grow in the future and wind turbine research will be necessary for improvements in the industry.

Development of the wind turbine technology requires more knowledge about the turbine performance and the process of power extraction. Special interest has raised for the wind tur-

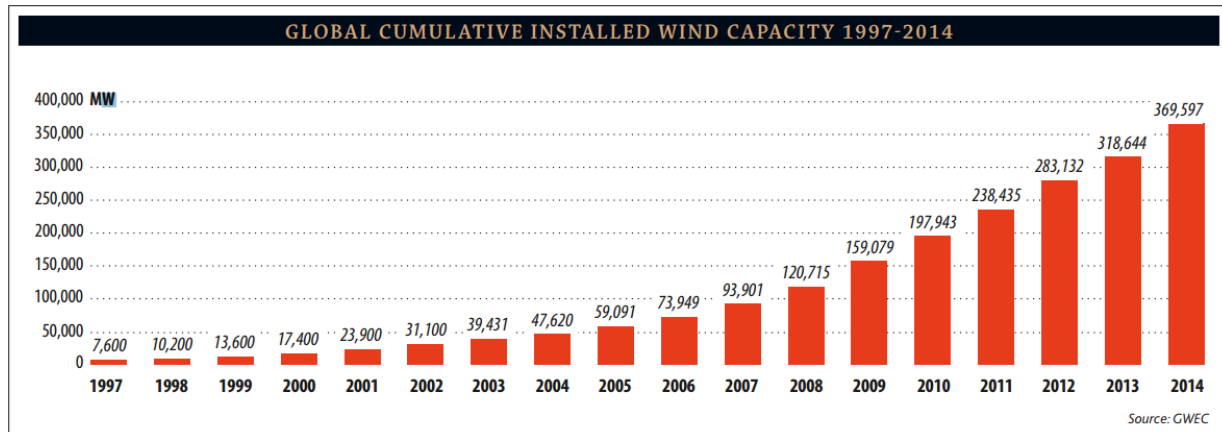
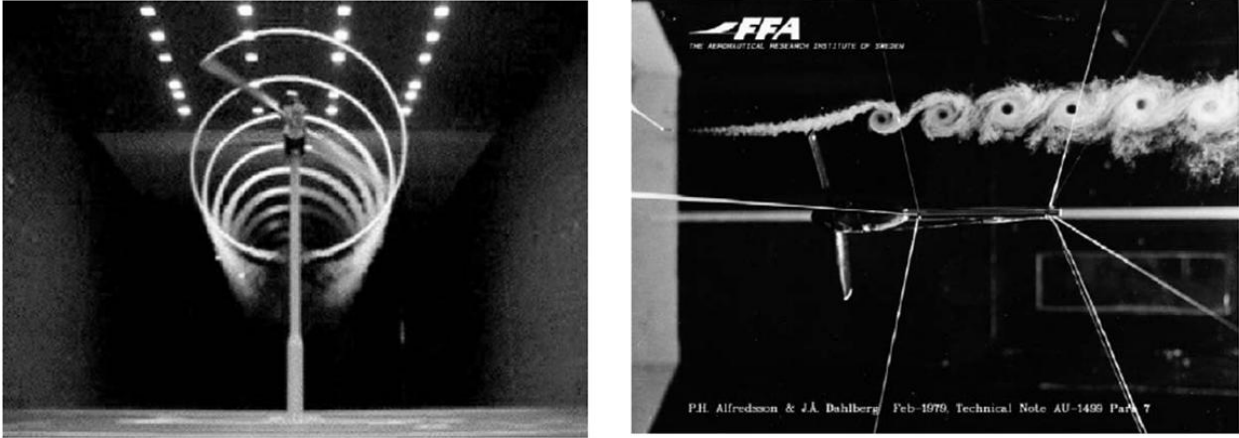


Figure 1.1: Global cumulative installed wind capacity from 1997 to 2014 [24]

bine wake due to its complexity and connections to the turbine's power production, and its disturbance to downstream turbines in wind farms. The wind turbine wake is the phenomena that occurs when wind exits the turbine blades. Counter rotating helical vortex pairs, consisting of a tip and root vortex shed from each blade of the turbine [20]. These structures, seen from figure 1.2, affects the wake directly downstream of the turbine blades, and are known as the near wake.

Full scale wind turbines are located in the lower region of the atmospheric boundary layer (ABL), where turbulence is present [23]. This makes the wind conditions unstable, and relying on a constant velocity is therefore not an option. The turbulent conditions will influence the wake evolution downstream, and are of special interest in wind farms. The wake can influence the power production of the downstream turbines, and abbreviate their fatigue life.

The research on wind turbine aerodynamics has had strong influence on the success of the modern wind energy. However, the engineering rules used might have limited applicability to special conditions, and a more physical understanding is needed [27]. Furthermore, the extensive use of the two-dimensional blade design methods (eg. XFOIL code) may omit important three-dimensional effects that can have strong influence on the turbine performance. Due to these issues, among other, a more fundamental approach based on experiments in controlled conditions are governing more of the modern research programs [27].



(a) Visualization of downstream evolution of helical wake [1] (b) Visualization of turbine wing tip vortices [1]

Figure 1.2: Wake structures [1]

1.2 Problem formulation

Performing experiments requires knowledge about different experimental methods and their corresponding accuracies. There may be several ways to perform tests, but being able to validate how accurate the results will become, can be crucial in getting a trustworthy conclusion.

A wind turbine experiment in the Fluids laboratory at NTNU was to be performed in this thesis. The construction of the experimental facility did not finish in time to be used in this thesis, but experiments will be performed in the near future. The experimental facility involves a simplified turbine wing being rotated and translated through a large tank of water. Water is used in order to obtain equal Reynolds number¹ as for full scale turbines, at lower velocities. This can be obtained due to the density and viscosity difference of water and air. The goal of the experiment is to visualize the complexity of the wing tip vortices and see how the wake is affected and develops downstream. An objective to be investigated in the rig, is how potential vibrations of the blades affects these vortices. To evaluate if vibrations occur, the trajectory of the blades needs to be tracked and reconstructed. This leads to the work to be conducted in this thesis, now being preparation work for the water tank experiment.

The aim of this thesis is to reconstruct trajectories of a rotating object by tracking the path of a point on the object. Estimating the accuracy of the tracking is also necessary to determine

¹The ratio of inertial forces to viscous forces

how well the potential blade vibrations can be determined. Cameras will be used as measuring apparatus. This involves the cameras being modelled and calibrated in order to correlate 2D image frames to the 3D world space. Camera modelling is a mathematical procedure, and will be elaborated in chapter 2. Accuracy of camera calibration is an important factor of how precisely a point can be tracked. Experiments and numerical investigation is to be done to evaluate the accuracy of the calibration and tracking. Additionally, it is also of special interest to study positioning of the cameras, and how it is related to accuracy of tracking.

Objectives of this thesis is therefore

- Determine accuracy of experimental camera calibration. I.e. determine expected reconstruction errors.
- Determine optimal camera position in space relative to the object to be tracked. This involves finding optimal number of cameras and angle between them resulting in smallest reconstruction error.
- Determine accuracy of experimental point tracking and reconstruction of object trajectories.

1.2.1 Previous work

Much of the theory behind previous work are complex and needs understanding before its correlations to the thesis' aim may be clear. The reader is therefore recommended to look into the theory in chapter 2 for a better understanding of the previous work presented in the next paragraphs.

The first objective is to determine the accuracy of the camera calibration. Different methods for calibrating cameras have been established by Tsai [26], Zhang [30],[18] and Faugeras [3]. All of the above suggests calibration with both 2D and 3D calibration objects, however, Zhang [18] points at the highest accuracy being obtained using a 3D apparatus. Of the Tsai [26] calibration method, accuracy study of using 2D object and 3D object has been performed by Samper et al. [22]. They found it possible to reach results from a 2D calibration object close to those using 3D objects as long as orientation, position and choice of correct system are maintained. Samper et

al. also states that the parameters of depth are the most sensitive parameters, and must be paid special attention to when calculated.

Continuing with the positioning of the cameras to optimize the tracking accuracy. Earlier experimental and theoretical research on optimal angles between cameras for particle tracking and particle recognition can be found for both particle image velocimetry (PIV) and particle tracking velocimetry (PTV) analysis in Maas [12] and Lawson et al. [11], [10]. Both PIV and PTV requires particle recognition by cameras, implying that analysis from both methods can be used as validation data. Maas [12] does not provide an optimal angle between the cameras, but states that at least three, optimal four, cameras should be used to decrease the number of ambiguities related to detecting and tracking particles. Lawson et al. [10] performed an experimental analysis of a stereoscopic² camera setup with varying angles between the two cameras in experimental PIV. Their results yield an optimal angle α between the cameras and z -axis of $20^\circ < \alpha < 30^\circ$, which contradicts with the theoretical values presented in their first paper [11] being 45° . Guezennec et al. [5] uses an orthogonal setup of the cameras for PTV, implying a 45° between each camera and the z -axis being optimal. The latter however, used computer generated images which did not include the aberrations connected to camera geometry and imaging in PIV test blocks [10].

1.3 Thesis outline

The paper is presented as a technical report. Primarily, the theory is presented, followed by the methods used to perform the numerical and experimental analysis. The results are presented both illustrative and tabulated together with corresponding discussions in chapter 4. Both chapter 3 and 4 will have sections and subsections dividing the numerical studies from the experimental studies. Finally, a conclusion of the whole project is given together with suggestions for further work. The latter will include information about the water tank experiment introduced in section 1.2.

²A setup with two cameras

Chapter 2

Theory

The aim of this thesis is to be able to determine the path of a rotating object by using cameras as measuring equipment. Performing this, cameras need to capture frames of the object during a transient sequence. Two dimensional image coordinates can, through a process known as triangulation, be back projected into three dimensional coordinates in space. By applying a recognisable and circular point to the object, its accurate center point can be determined. If the same point in two or more following frames are recognised, the path of the point can be determined. This procedure is explained more thoroughly in section 2.3. Information about the camera characteristics is necessary in order to conduct a triangulation, and is obtained through a camera calibration. Camera calibration is the process of using known information in a world reference frame to obtain information about a corresponding two dimensional image frame. The procedure for calibration is elaborated in section 2.2. How the 2D image space is related to the 3D world space, is described by a camera model. Hence, a camera model consists of parameters that are determined through a camera calibration [21].

The chapter starts with introducing a camera model, which is the most important topic in order to understand the other mentioned techniques. Thereafter follow calibration techniques, back projection of world coordinates, and tracking. At the end of this chapter, some information about the apparatus and electronics used in the experiments and simulations conducted is provided, including types of cameras and different software used.

2.1 Camera modelling

Camera calibration is a necessary process in order to extract geometric information from two dimensional images [18]. This means for instance being able to evaluate distances between, and positions of, objects in the 3D space from a 2D image frame. Finding the relationship between an image frame and the world space, a camera can be modelled mathematically. Before a calibration can be conducted, it is important to understand how the camera is modelled and how the camera parameters in the model are connected. The most common and simple camera model is based on the so called pinhole camera illustrated in figure 2.2, and is introduced in the next section. Other camera models are for instance the fish-eye camera model depicted in figure 2.1a and generic camera model in figure 2.1b [21], these will not be explained further in this thesis.

2.1.1 Pinhole camera model and perspective projection

The pinhole model is the simplest camera model there is and is depicted in figure 2.2. The pinhole camera model is a fairly accurate model for the geometry and optics in most modern Vidicon, CMOS (Complimentary metal-oxide semiconductors), CCD (Charge coupled device) and CID (Charge injection device) cameras [3]. CMOS cameras are used in the experimental analysis in this thesis, and the theory of the pinhole model is therefore a suitable choice for modelling the camera. Most of the theory given related to this model is gathered from Faugeras

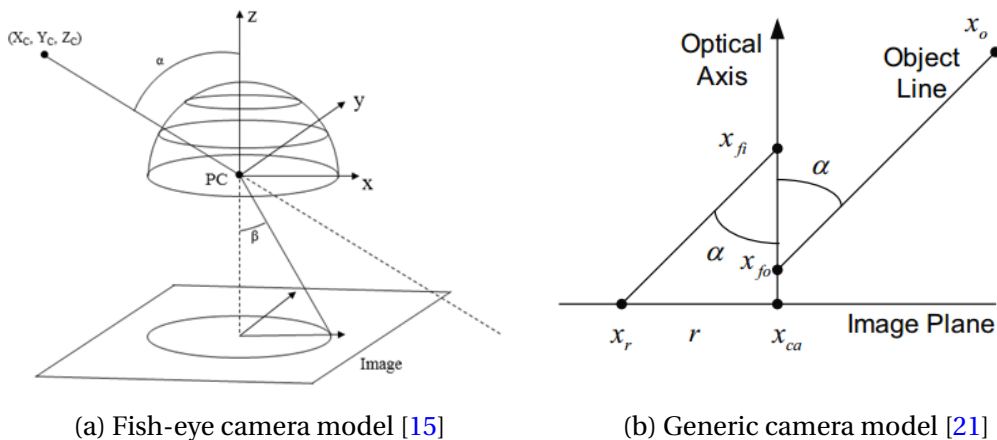


Figure 2.1: Illustrations of additional camera models

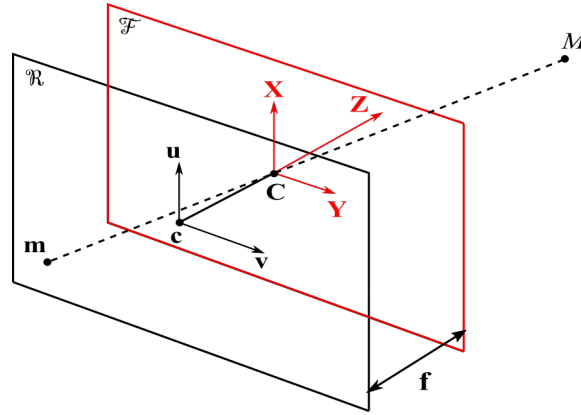


Figure 2.2: Pinhole camera model

[3].

With reference in figure 2.2, it consists of a retinal plane \mathcal{R} , which the image is formed on through a perspective projection. This operation happens through a point C , being the optical center: An image \mathbf{m} is formed at the retinal plane of the three-dimensional point M , through the optical center illustrated by the dashed line from \mathbf{m} to M in figure 2.2. The distance f between the principal point C and the retinal plane is known as the focal length of the optical system. The image point \mathbf{m} gets the position in the retinal plane where the line $\langle C, M \rangle$ intersects the plane [3]. The plane going through camera center C is called the focal plane \mathcal{F} , and is parallel to \mathcal{R} . The line perpendicular to \mathcal{F} and intersects C , is called the optical axis, and intersects the retinal plane at \mathbf{c} .

With these parameters known, we can look further into to the geometry: the coordinate system of the three-dimensional space is defined as (C, X, Y, Z) ; and the retinal coordinate system as (c, u, v) . Following, the relationship between the image coordinates and 3D world coordinates can be written as

$$-\frac{f}{Z} = \frac{u}{X} = \frac{v}{Y} \quad (2.1)$$

This relationship can be rewritten linearly as

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.2)$$

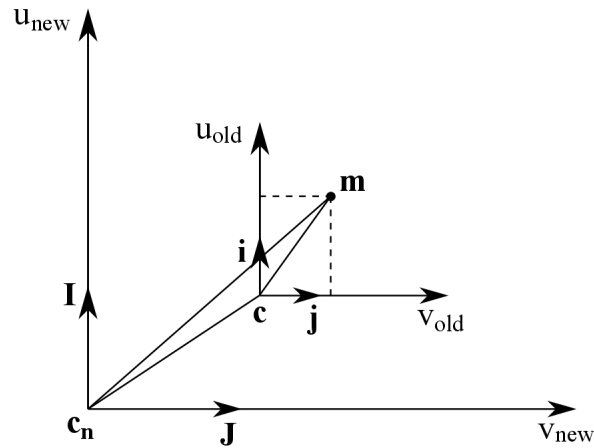


Figure 2.3: Change of coordinate system in the retinal plane

where $u = \frac{U}{S}$ and $v = \frac{V}{S}$ if $S \neq 0$ [3]. $S = 0$ is the equation of the line at infinity of the retinal plane, meaning the world point is located at the focal plane. The relationship between the projective image- and 3D-world coordinates given in 2.2 is linear and can be written in matrix form as

$$\tilde{\mathbf{m}} = \tilde{\mathbf{P}}\tilde{\mathbf{M}} \quad (2.3)$$

where $\tilde{\mathbf{M}} = [X, Y, Z, 1]^T$, $\tilde{\mathbf{m}} = [U, V, S]^T$ are the augmented or homogeneous representation of $M = [X, Y, Z]$ and $\mathbf{m} = [u, v]$. Homogeneous coordinates are a representation where each point in \mathbb{R}^n are associated with a line in \mathbb{R}^{n+1} , a line that passes through origin [9]. $\tilde{\mathbf{P}}$ is the perspective projection matrix, consisting of all camera parameters to be elaborate din next section. Equation 2.3 implies how the camera system can be considered to perform a linear projective transformation from the projective space \mathcal{P}^3 into the projective plane \mathcal{P}^2 . By introducing this linear relation, the power of linear algebra can be applied instead of dealing with nonlinear equations [3]. Matrix $\tilde{\mathbf{P}}$ can be written as $[\mathbf{P} \ \tilde{\mathbf{p}}]$, and can then be used to obtain useful information about the camera center \mathbf{C} and the the optical ray being the extension of $\langle \mathbf{C}, \mathbf{m} \rangle$. This will be described later in this chapter.

2.1.2 Intrinsic and extrinsic camera parameters

To define intrinsic (internal) and extrinsic (external) parameters, a change of coordinate system is first necessary.

In camera calibration, the coordinate system of the image frame is usually centred in the corner of the image. As described earlier, and illustrated in figure 2.2, the center is located at point \mathbf{c} as the center of the retinal plane. Figure 2.3 shows how the change of coordinate systems happen in the retinal plane, changing the center from \mathbf{c} to \mathbf{c}_n . The new path towards the image point \mathbf{m} is now $\mathbf{c}_n - \mathbf{c} - \mathbf{m}$. Introducing a scaling $\mathbf{s} = \begin{bmatrix} k_u & 0 \\ 0 & k_v \end{bmatrix}$ from \mathbf{i} and \mathbf{j} , to \mathbf{I} and \mathbf{J} , and letting \mathbf{t} be the translation from \mathbf{c}_n to \mathbf{c} , the projection matrix becomes

$$\tilde{\mathbf{P}} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.4)$$

where u_0 and v_0 are the parameters of the principal point \mathbf{c} , and k_u and k_v the scaling factors expressed in pixels per meter. $-fk_u$ and $-fk_v$ can be interpreted as the size of the focal length in horizontal and vertical pixels, respectively [3]. Neither $-fk_u$, $-fk_v$, u_0 or v_0 depends on the location of the camera in space, and are known as the intrinsic parameters. The first 3×3 matrix of equation 2.4 is therefore known as the intrinsic matrix, and is often written as

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

where $\alpha = -fk_u$, $\beta = -fk_v$, and γ is the skew which equals zero for quadratic pixels. The quantities u and v can now be written in pixel values as

$$u = \frac{U}{S} = \alpha \frac{X}{Z} + u_0 \quad (2.6)$$

$$v = \frac{V}{S} = -\beta \frac{Y}{Z} + v_0 \quad (2.7)$$

As changing the retinal coordinate system gives us the intrinsic camera parameters, the extrinsic parameters are obtained by changing the world reference frame: the 3D coordinate system. The extrinsic parameters indicate the position and the orientation of the camera in a world reference frame, and can be defined mathematically by the camera center \mathbf{C} and a 3×3 rotation matrix \mathbf{R} [19]. \mathbf{R} consists of three individual rotation matrices where each determines a rotation

about an axis. The three rotations are known as *roll*, *pitch* and *yaw*, and are rotations about x , y and z -axis, respectively [4]. They are defined as

$$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) \\ 0 & \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \quad (2.8)$$

$$R_{pitch} = \begin{bmatrix} \cos(\theta_j) & 0 & \sin(\theta_j) \\ 0 & 1 & 0 \\ -\sin(\theta_j) & 0 & \cos(\theta_j) \end{bmatrix} \quad (2.9)$$

$$R_{yaw} = \begin{bmatrix} \cos(\theta_k) & -\sin(\theta_k) & 0 \\ \sin(\theta_k) & \cos(\theta_k) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

where θ_i , θ_j and θ_k are the Euler angles about x , y and z axis respectively. \mathbf{R} is an orthogonal matrix, i.e. $\mathbf{R}^t = \mathbf{R}^{-1}$, and it follows that $\det \mathbf{R} = \pm 1$.

To find the new projection matrix including the extrinsic parameters, the same procedure as for the retinal coordinate system can be used. The aim is to find the path \mathbf{CM} which now

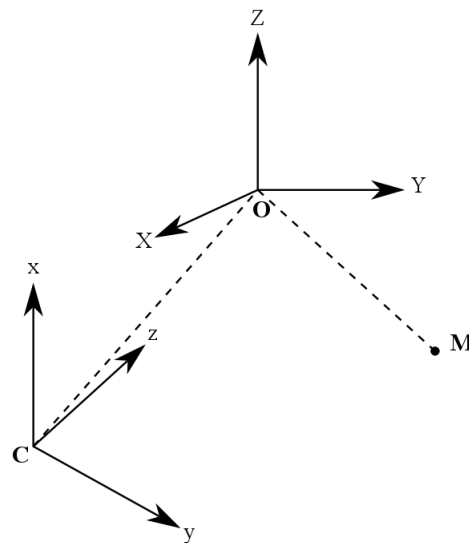


Figure 2.4: Change of world reference frame

equals $\mathbf{CO} + \mathbf{OM}$. The vector \mathbf{CO} is denoted as a translation $\mathbf{t} = [t_x, t_y, t_z]^T$, while \mathbf{OM} is the new coordinate values rotated from the old system to the new system depicted in figure 2.4. \mathbf{CM} can be written

$$\mathbf{CM} = \mathbf{t} + \mathbf{R} \begin{bmatrix} x_{new} \\ y_{new} \\ z_{new} \end{bmatrix} \quad (2.11)$$

The projection matrix is therefore multiplied with the 4×4 matrix

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix}$$

where \mathbf{R} and \mathbf{t} are the extrinsic parameters of the camera describing the orientation and position of the camera, respectively, with respect to the new world coordinate system. The updated perspective projection matrix can be found by multiplying matrix $\tilde{\mathbf{K}}$ on the right side of the old perspective projection matrix. Using the old matrix $\tilde{\mathbf{P}}$ given in equation 2.4, the general form of $\tilde{\mathbf{P}}$ can be expressed as a function of both the intrinsic and the extrinsic parameters as

$$\tilde{\mathbf{P}} = \begin{bmatrix} -\alpha \mathbf{r}_1 + u_0 \mathbf{r}_3 & \alpha t_x + u_0 t_z \\ \beta \mathbf{r}_2 + v_0 \mathbf{r}_3 & \beta t_y + v_0 t_z \\ \mathbf{r}_3 & t_z \end{bmatrix} \quad (2.12)$$

or, written in matrix form as

$$\tilde{\mathbf{P}} = \mathbf{A} [\mathbf{R} \ \mathbf{t}] \quad (2.13)$$

2.1.3 Lens distortion

The pinhole camera model is collinear and has no distortion, meaning the straight line in object space is mapped to a straight line in the image frame. Usually, a lens system causes some distortion that contributes to a difference between the real image and the pinhole image [21].

Lens distortion can be divided into two types: radial distortion; and decentering distortion. Radial distortion is symmetric and ideal image points are distorted along radial directions from



Figure 2.5: Example of lens distortion of object with the distorted image to the left and undistorted image to the right [8]

the distorted center [18]. It is caused by imperfect lens shape, and can be as illustrated in figure 2.5. Decentering distortion is usually caused by improper lens assembly, and ideal image points are distorted in both radial and tangential directions [18].

Zhang [18] introduces the steps of dealing with lens distortion as:

1. Lens distortion gives the true image coordinates by adding the distortion to u and v provided in equation 2.1. The new \check{u} and \check{v} coordinates becomes

$$\check{u} = -f \frac{X}{Z} + \delta_x$$

$$\check{v} = -f \frac{Y}{Z} + \delta_y$$

where δ_x and δ_y are the distortions.

2. Secondly, the affine transformation from the true image coordinates to pixel coordinates given in equations 2.6 and 2.7, becomes u_{dist} and v_{dist} with \check{u} and \check{v} :

$$u_{dist} = k_u \check{u} + u_0$$

$$v_{dist} = k_v \check{v} + v_0$$

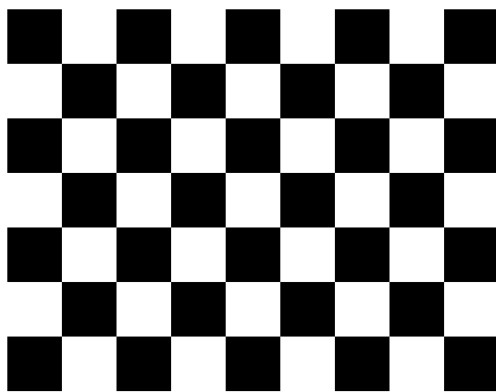
The interested reader is referred to Zhang [18] for more elaborated information about lens distortion and how the parameters are derived.

2.2 Camera Calibration

Up to now, the correlations between the camera parameters of a pinhole model are known. A camera can then be calibrated to estimate the extrinsic and intrinsic camera parameters of the model. These parameters are necessary to correlate the 2D image coordinates \mathbf{m} to the 3D world coordinates M . The calibration process involves capturing images of a known pattern, to use the known dimensions in world space to calculate the camera parameters of the model [21], described in previous section. Both 3D and 2D calibration objects can be used in this process, consisting of a recognizable pattern where dimensions and shape are known [18]. The most accessible object for 2D calibration is a checkerboard illustrated in figure 2.6a of white and black squares. The corners of the board are easy to detect, and it is cheap to produce. Other typical objects may consist of circle points, have variable depths, or consist of orthogonal planes. An example of a 3D calibration object is depicted in figure 2.6b.

The 3D world points are defined with reference to the calibration object. From the image frames, the corresponding 2D reference points are found, and a fit is performed. The image frame consists of pixels determined by the resolution of the camera lens. For example, a frame consisting of 1024x1024 pixels is a matrix of indices from 1 to 1024 in both lateral and longitudinal direction, having same color within each index.

Zhang [18] suggests different calibration approaches: self-calibration, 1D line based, 2D



(a) 2D checkerboard calibration object



(b) 3D calibration object from [18]

Figure 2.6: Examples of 2.6a 2D and 2.6b 3D calibration objects

plane based, and 3D reference object calibration. Due to the necessity of good accuracy, only 2D and 3D apparatus calibration will be presented. The two different approaches of performing a camera calibration is therefore: 1) a 2D approach where all the points at the calibration object are assumed to be located at $Z = 0$, and 2) a 3D approach where the depth of the different points are perfectly known.

2.2.1 2D calibration

The 2D calibration procedure is summarized by the flow chart below, figure 2.7. Multiple images are first captured of the plane calibration object from different orientations relative to the camera. This is necessary to determine the correct intrinsic parameters of the camera. The calibration object does not necessarily need to cover the whole image frame, though it should be moved around in the image frame to capture the radial distortion of the image [26], [18]. The extrinsic parameters: the rotation and position of the camera, will obviously vary from image to image with different orientations. That is due to the xy -plane always corresponds with the plane of the checkerboard. Since all calibration points are assumed to be located in plane $Z = 0$, one can substitute the actual projection matrix by an homography \mathbf{H} . \mathbf{H} omit the last row of the rotation matrix, \mathbf{r}_3 and of the translation vector \mathbf{t} , t_3 . The different parameters of \mathbf{H} is denoted h_i .

The 2D object calibration method is applied in the experimental work to be conducted in this thesis. The reason for this is the flexibility of this method in terms of cost and handling of a checkerboard.

2.2.2 3D calibration

Calibrating a camera using a 3D object does not require multiple images to be captured. The 3D calibration object on the other hand, must either be angled or have different depths. Cubes with checkerboard patterns on each sides are common, where the image frame captures three sides of the cube. It can also be done using a plane object. Capturing images of the plate at known and different displacements will also result in a 3D calibration. The procedure for the 3D camera calibration can be summarized by the steps showed in figure 2.8, and are based on the

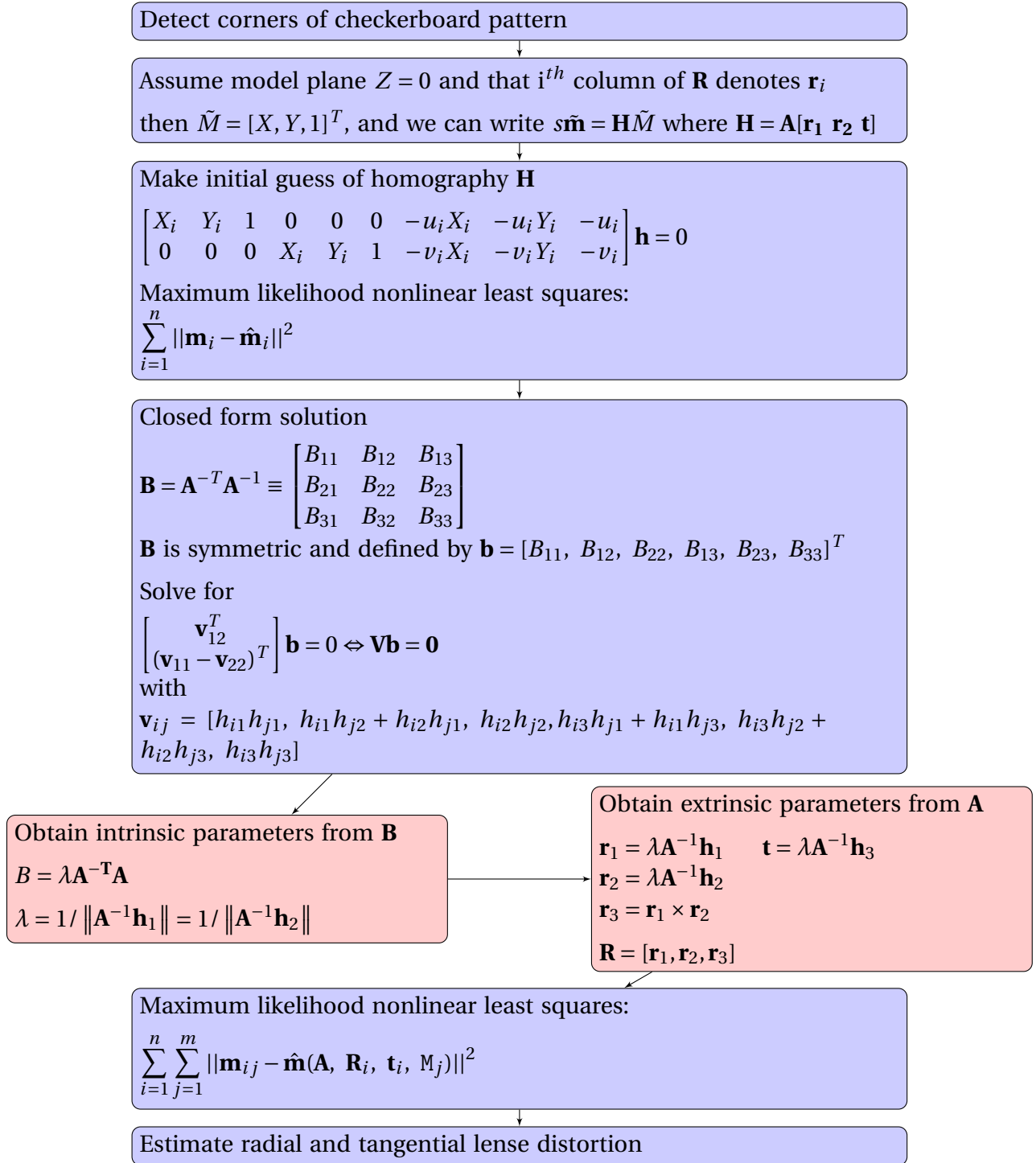


Figure 2.7: Flow chart of 2D camera calibration

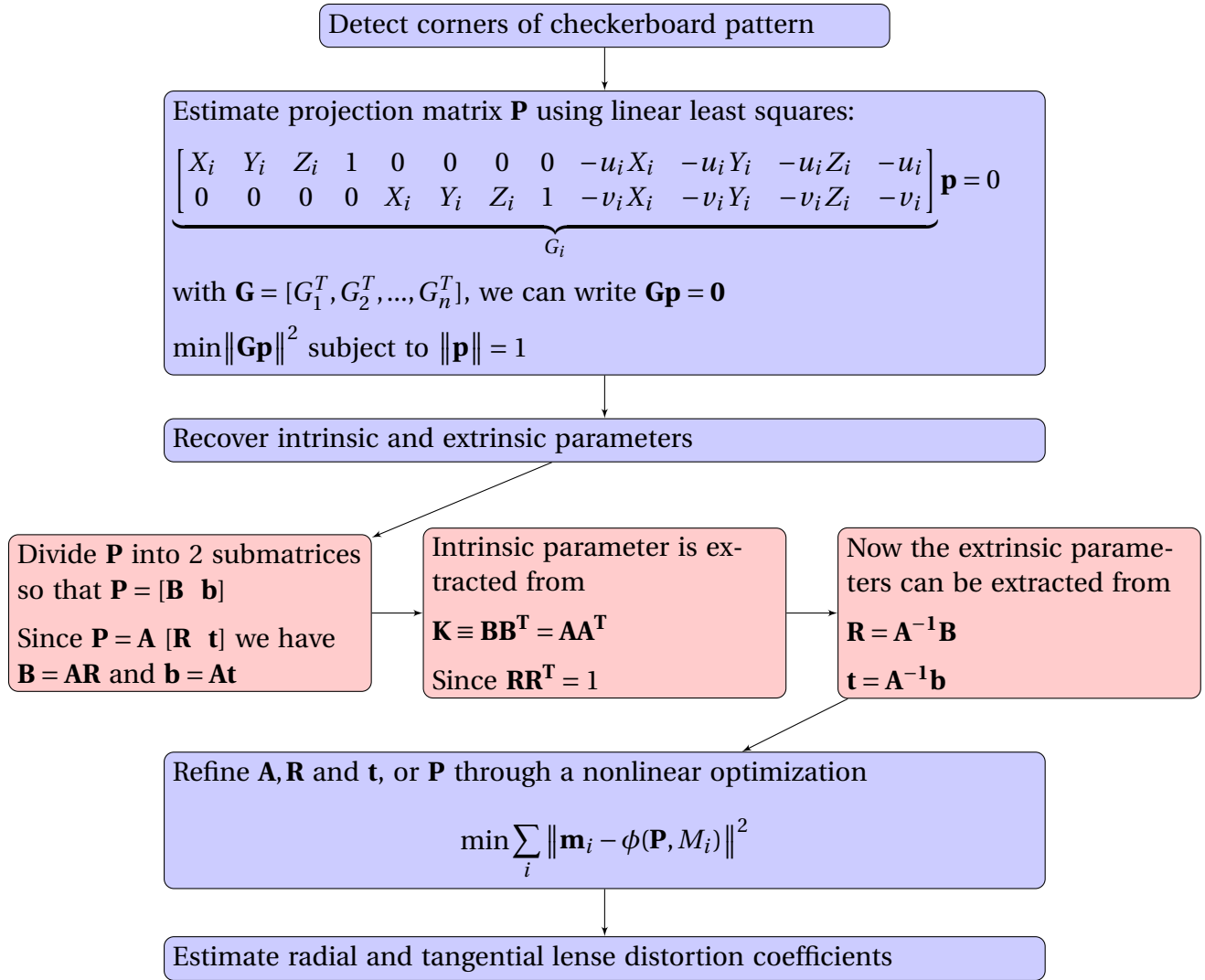


Figure 2.8: Flow chart of 3D calibration steps

procedures of Faugeras [3] and Zhang [18].

2.3 Reconstruct world position

After a camera is calibrated and the parameters are obtained, the 2D image coordinates can be back projected into 3D world coordinates through a process called triangulation. From the name triangulation, it might be understood that three objects, at a minimum, are necessary. The three objects are here referred to as two cameras and one object. Looking at figure 2.9, the

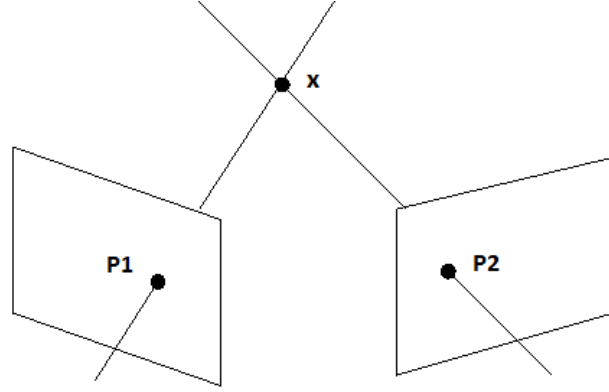


Figure 2.9: Illustration of triangulation with two cameras: the same point, \mathbf{x} is identified in frames from two cameras as $\mathbf{P1}$ and $\mathbf{P2}$.

same point \mathbf{x} are projected onto frames of both cameras as image coordinates $\mathbf{P1}$ and $\mathbf{P2}$.

Recalling equation 2.3, the pixel position $\tilde{\mathbf{m}}$ is obtained from the world coordinate position $\tilde{\mathbf{M}}$ multiplied with the projection matrix. This relationship holds up to a scale factor λ , and can be written as

$$\lambda \tilde{\mathbf{m}} = \tilde{\mathbf{P}} \tilde{\mathbf{M}} = \mathbf{A} [\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{M}} \quad (2.14)$$

Rearranging the equation and using the rules of matrix calculations, the back projection of a 3D world point can be written as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{C} + \lambda \mathbf{R}^{-1} \mathbf{A}^{-1} \tilde{\mathbf{m}} \quad (2.15)$$

where $\lambda = \frac{Z - C_3}{z_3}$ and $(z_1, z_2, z_3)^T = \mathbf{R}^{-1} \mathbf{A}^{-1} \tilde{\mathbf{m}}$ [19]. The step by step calculation from equation 2.14 to 2.15 can be found in Appendix section A.1.

The expression on the right hand side corresponds to an optical ray, also known as the line of sight from the camera lens, going through point $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$ from the camera center \mathbf{C} illustrated in figure 2.9. Having knowledge about the depth is therefore required up to a certain extent, in order to calculate a long enough line (ref. Z in equation for λ). In order to obtain the world position of one point in corresponding images, this ray must be calculated for all cameras used in the process, with their individual corresponding calibration parameters. The intersection point of the optical rays from the cameras, will result in the exact 3D world position.

However, due to calibration errors and random errors due to image noise, corresponding

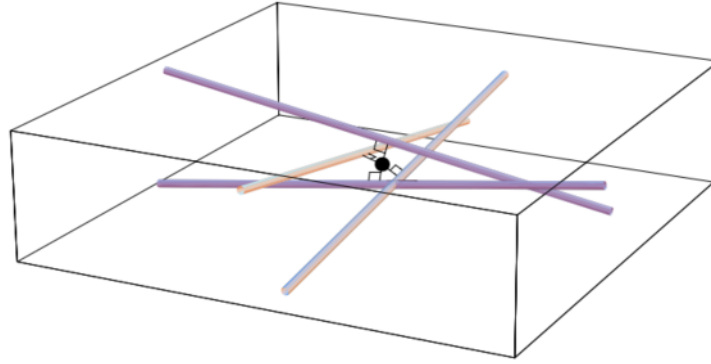


Figure 2.10: Illustration of closest intersection point of almost intersecting lines of sight, from Mann and Ott [13]

lines of sight from cameras does not intersect. They are usually very close, but will never cross each other perfectly. A reasonable assumption which solves this problem, is assuming the point is located at the closest intersection point of all lines. An algorithm finding this intersection point of 3D lines are proposed by Mann and Ott [13], and relies on finding the minimum of the sum of the squared distances S

$$S = \sum_{i=1}^n \text{dist}(\mathbf{p}, l_i)^2 \quad (2.16)$$

where $\text{dist}(\mathbf{p}, l_i)$ is the distance between the point \mathbf{p} and line l_i illustrated in figure 2.10. The point is found by introduce a parametric form for l_i : $t \rightarrow \mathbf{a} + \mathbf{b}t$, with $\|\mathbf{b}\| = 1$. $\text{dist}(\mathbf{p}, l_i)$ is the length of the projection $\mathbf{p} - \mathbf{a}$ on to the plane perpendicular to \mathbf{b} , hence

$$\text{dist}(\mathbf{p}, l - i) = \|P_b(\mathbf{p} - \mathbf{a})\| \quad \text{with} \quad P_b = I - \mathbf{b} \otimes \mathbf{b} \quad (2.17)$$

P_b can also be expressed with indices as $P_{b,ij} = \delta_{ij} - b_i b_j$. To find the minimum of the squared distance, the gradient of the squared equation 2.17 must be set to zero. It can easily be shown that $\nabla_{\mathbf{p}} \text{dist}(\mathbf{p}, l_i) = 2P_b(\mathbf{p} - \mathbf{a})$ implying that

$$\begin{aligned} \nabla_{\mathbf{p}} S = 0 &\implies \sum_{i=1}^n P_{\mathbf{b}_i} \mathbf{p} = \sum_{i=1}^n P_{\mathbf{b}_i} \mathbf{a}_i \\ \mathbf{p} &= \left(\sum_{i=1}^n P_{\mathbf{b}_i} \right)^{-1} \sum_{i=1}^n P_{\mathbf{b}_i} \mathbf{a}_i \end{aligned} \quad (2.18)$$

Substituting this value into equation 2.16, gives the minimum value for S [13].

2.4 Point Tracking

The aim of the thesis is to reconstruct the path of a rotating object in space. Procedures of point tracking are therefore elaborated in this section in order to obtain the trajectory of a moving object. There are many ways to track points in space depending on size, shape and quantity of points to be tracked at a time. Point tracking and particle tracking are used in different applications. In fluids engineering particle tracking velocimetry (PTV) is a very common procedure to measure the three dimensional velocity [12]. Here, many small particles are injected to the flow, illuminated and tracked. They can then be recognised in images through their higher or lower intensity than the flow background [12]. Motion tracking in film making is also a very common procedure to transfer human movements over to animated movie figures [29]. There are a number of different technologies for tracking human motion: magnetic trackers, optical tracking systems and mechanical tracking systems, to name a few [29].

The tracking procedure of PTV is very close to the procedure to be used in this thesis. It can be divided into four steps, where most of them are already described in earlier sections. However, a brief repetition will be given in order to connect the steps together into the tracking procedure.

Step 1

The first step in the procedure is calibration of the cameras to establish the camera characteristics. These are necessary to evaluate the 3D particle position from 2D image pixel coordinates. The procedure of evaluating these are discussed more thoroughly in the theory section 2.2 about camera calibration. The typical installation in 3D PTV is three or four cameras positioned in an angular configuration. Two cameras may also be used, but the accuracy decreases significantly from three to two cameras, see Maas[12]. More than four cameras will rarely be necessary.

Step 2

At least two cameras must be used to capture frames of each timestep before the particles can be identified [6]. Point or particle identification, can be determined by scanning through the intensity of each pixel, or determine the point through shape recognition. Performing the particle

intensity scan, a threshold is set for intensity, and pixels above the certain threshold is registered. After the search is completed, the size and centroid of the particles can be determined, as well as those that are overlapping [6]. Performing the shape detection method, recognising circular objects can be done with implemented circle detection algorithm in MATLAB¹. The center point and size of the detected circles are then returned, and can be used further in the tracking process.

Step 3

Reconstruction of the 3D particle or point position is the next step. Having found the center of a point in the frame from one camera, the same point can be found in the frame of the another camera(s). Calculating the lines of sight from all cameras, the three dimensional coordinates of the point can be determined by finding the closest intersection point of the lines (see section 2.3). After step three, one list of three dimensional point coordinates are obtained from $N \times$ cameras lists of 2D image coordinates [6].

Step 4

The last step is particle tracking, which requires two sets of 3D coordinates of particles captured at time t and $t + \Delta t$. For this to be done correctly, Step 2 is important in order to detect the same point or particle in image $t + \Delta t$ as in frame t [6]. A search window, or a constraint for how the algorithm detects the point is often necessary. Having correct points from image t and $t + \Delta t$, a 3D displacement plot can be obtained.

To detect circular points in an image, MATLAB has a function `imfindcircles` [17], found in the Image Processing Toolbox. The function is able to detect circular object in a gray scale image using the Hough transform [7]. Radii and center values of the detected circles are returned as vector and matrix forms, respectively. Different parameters can be changed to enhance the detection of the circles[17], being circle radius span, object polarity, sensitivity, computation method, and edge threshold. An elaborated description of how these parameters should be set are provided in Appendix section A.2.

¹MATLAB is a numerical computer software used for different purposes within among other computer vision and image processing



(a) Photron FASTCAM sa1.1



(b) Nikon AF micro Nikkar 60mm lens

Figure 2.11: Camera and lens used in experiments

2.5 Apparatus and Electronics

In this section the different equipment and devices used in the experiment are presented. For calibration of cameras, post processing of results and investigating accuracies numerically, MATLAB R2015b is used together with special toolboxes as Computer Vision Toolbox.

For the camera calibration and tracking experiment, two Photron FASTCAM SA1.1 high-speed cameras as showed in figure 2.11a are used. The cameras can capture up to 5000 frames per second (fps) at one mega pixel resolution. To capture and process the images, Photron FASTCAM Viewer (PFV) is installed on the computer and connected to the camera setup. From this software, variables as fps, image resolution and triggering options can be selected.

The lenses used in both the calibration and point tracking experiment depicted in figure 2.11b, are of type Nikon AF micro Nikkar 60 mm 1:2.8 D.

Moving the tracking object is done by a SMSD-822 step motor controller from Smart Motor Devices (see manual [25]). The controller responds on step signals, and is therefore connected to a signal generator of type digimess FG200 2MHz.

Chapter 3

Methodology

To reach the desired objectives of investigation, different analyses needs to be performed to evaluate expected accuracy of tracking. In this chapter, the procedures of the experiments and numerical analyses conducted are described. Primarily, the method applied for calibration of the cameras is explained. This involves the approach for a camera calibration experiment, that is done to evaluate the uncertainty of back projection, and the calibration before point tracking. Secondly, the procedure behind the numerical analysis is presented. Here, the aim is to find the optimal camera positioning at which the experiment should be performed. Last, the description of the point tracking experiment is provided, which is conducted to estimate how well a path can be reconstructed of a rotating object.

3.1 Camera calibration

Both experiments in this thesis involves camera calibration. The method applied in the two experiments are equal, and follows the procedure of Zhang's [18] 2D-algorithm described in section 2.2.1. The algorithm is implemented in MATLAB Computer Vision Toolbox [16], and needs images captured of a checkerboard at different orientations, and the size of the squares on the checkerboard pattern object in mm. With this known, the algorithm calculates camera parameters with functions from the camera calibration application. The parameters are then processed further with functions for finding lines of sight and the camera center, a function for finding intersection point of lines, and functions to generate post processing plots. The code for

the calibration is provided in Appendix section C.1.

The checkerboard is of size 8x7 squares resulting in 42 calibration points, with square size of 25 mm, as depicted in figure 3.1. However, the tracking experiment calibrates the cameras using two different checkerboards: a 8x7 squares and 9x4 squares of size 25 mm and 24.5 mm, resulting in 42 and 24 calibration points, respectively. The approach of the calibration experiment is elaborated in the next section, while the methodology of the tracking is described later in this chapter, more specific in section 3.3.

3.1.1 Experimental camera calibration

To estimate expected reconstruction errors, a simple camera calibration experiment is conducted. The experimental setup consists of a Photron FASTCAM mounted onto a 100 cm long traverse, and a checkerboard calibration object described in previous section mounted on a small scaffold. The checkerboard is fastened to a hinge, which together with the scaffold can be rotated about two out of three axes. The object can be elevated and lowered to desired height. An image of the scaffold setup is depicted in figure 3.1.

Only one camera will be used in the experiment, but capturing images from different locations on the traverse will be done to simulate three camera views of the object. The camera locations on the traverse are at 0 cm (left end of traverse), 50 cm and 100 cm to obtain as wide

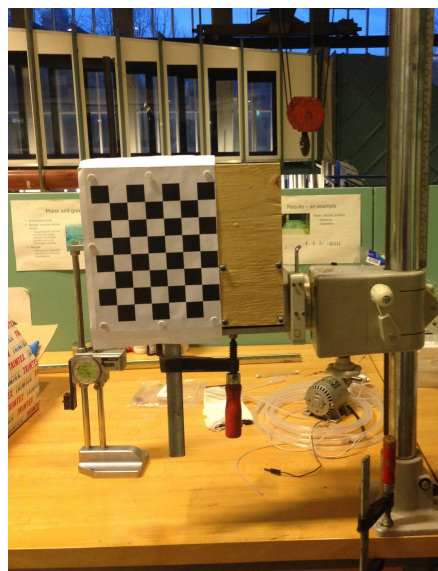


Figure 3.1: Object fastened to scaffold used for calibration

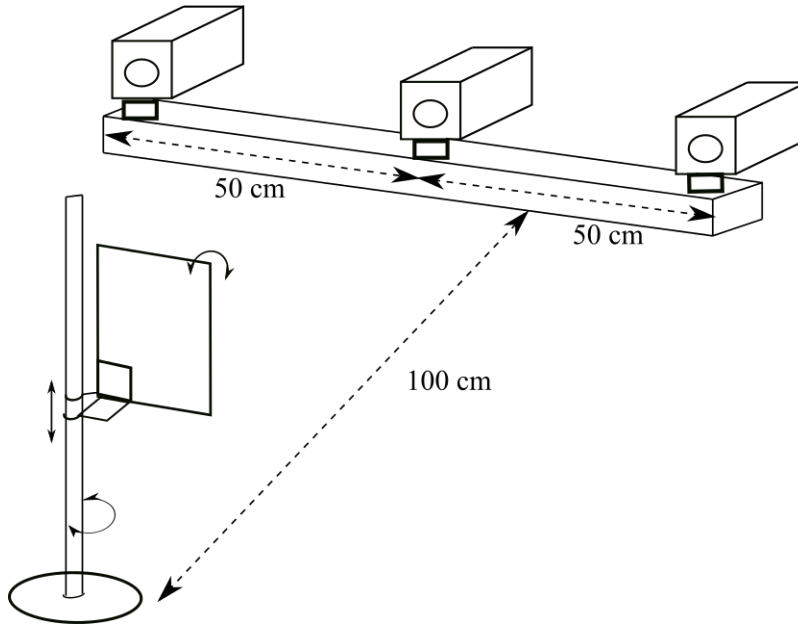


Figure 3.2: Experimental setup for the camera calibration

angles as possible between the cameras. An illustration of the setup is depicted in figure 3.2. The angle between the cameras are 20° , accounting for the lens being the center of the camera, located about 10 cm away from the edge of the traverse. The positions of the cameras and pitch, yaw and roll are summarized in table 3.1. The calibration object is located about 100 cm from the camera traverse.

In this calibration experiment, a number of 15 images will be captured and implemented in the algorithm. MATLAB Camera Calibration application suggests between 10 and 20 images for best results [16]. The choice of using 15 images is therefore sufficient in terms of accuracy and time consumption.

Table 3.1: Position and orientation of camera on traverse

	LEFT	MIDDLE	RIGHT
CAMERA ANGLE			
<i>Pitch</i>	2.5°	2.5°	2.5°
<i>Yaw</i>	25°	45°	65°
<i>Roll</i>	0°	0°	0°
LOCATION ON TRAVERSE	0 cm	50 cm	100 cm
REFERENCE POINT			
<i>x</i>	168 px	103 px	126 px
<i>y</i>	902 px	912 px	895 px

Table 3.2: Orientation of calibration object

ORDER	ORIENTATION
1	Base ^a
2	30° left
3	30° right
4	45° up
5	45° down
6	45° up, 30° left
7	45° down, 30° left
8	45° up, 30° right
9	45° down, 30° right
10	15° right
11	15° left
12	25° up
13	25° down
14	25° up, 15° left
15	25° down, 15° left

^a Base image plane is orthogonal to camera centerline

The 15 images are captured in two different procedures:

- (A) One object orientation is captured from all three camera positions, Before moving the object to the next position
- (B) All 15 object orientations are captured from one camera position at the time, before the same procedure is repeated for the remaining camera positions

The different object orientations of the calibration object can be found in table 3.2. The height of the hinge is not changed during the process.

3.2 Numerical accuracy analysis

Numerical analysis of camera positioning is performed to estimate how it affects the accuracy of back projection of world points. It is of interest to find the optimum angle between cameras of which the tracking experiment should be performed, and how many cameras that should be used. Different combinations of angles are therefore numerically tested with MATLAB using Monte Carlo simulations.

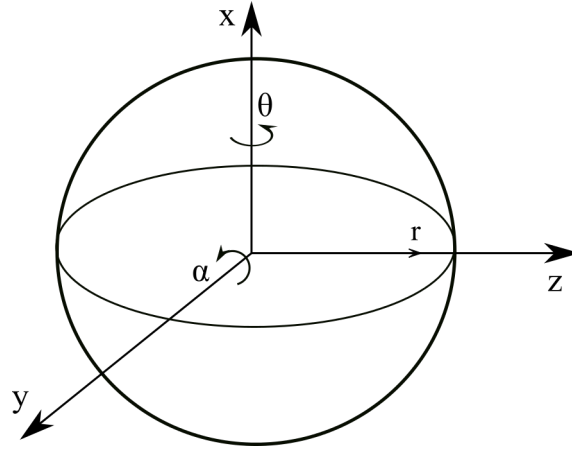


Figure 3.3: Illustration of spherical coordinates together cartesian coordinates [28]

To locate the cameras around the calibration object, spherical coordinates are most appropriate, where only the radius and plane angles are changeable variables. When estimating camera parameters, the cameras are located a depth or radius away from the object, and the checkerboard surface reflects the xy -plane. A natural choice of angles to vary therefore corresponds to a rotation of the xz -plane, α and yz -plane, θ . A representation of these customized spherical coordinates combined with cartesian coordinates can be seen in figure 3.3.

The code for the analysis is written in MATLAB and can be found in Appendix section C.2. Inputs to the analysis are following: an intrinsic camera matrix \mathbf{A} ; a radius or distance between the cameras to the object, r ; the angles α and θ ; and the number of cameras. Additionally, the rotation matrices are dependent on α and θ and must be changed from camera position to camera position. Recalling $\mathbf{R} = R_{yaw}R_{pitch}R_{roll}$, α is a *pitch* rotation, i.e. a rotation of the xz -plane about the y -axis, while θ is a *roll* rotation, implying a rotation of the yz -plane about

Table 3.3: Input values for numerical angle study for camera calibration accuracy

PARAMETER	VALUE
Intrinsic matrix	$A = \begin{bmatrix} 5386 & 0 & 494.6 \\ 0 & 5381 & 444.5 \\ 0 & 0 & 1 \end{bmatrix}$
Radius	$r = -1000$ [mm]
α	$10^\circ : 5 : 60^\circ$
β	$15^\circ : 15 : 60^\circ$
Number of cameras	2, 3 and 4

the x -axis. Therefore, when a camera is rotated in both *pitch* and *yaw* directions, the rotation matrix of that particular camera will depend on both angles, contra only one angle if it is only rotated in one direction. Similarly, the camera positions \mathbf{C} depends on these angles and must be changed for each camera setup. All of the different rotation matrices and camera positions set for the numerical analysis are provided in Appendix, section B.

The intrinsic matrix applied is chosen to be constant for all the camera variations, and have values equal to one matrix from the calibration experiment described in previous section. Choosing this matrix is reasonable when simulating equal radial distance to the calibration object from the camera lens. Fictive world points are then created, to calculate corresponding image points. Both α and θ will obviously vary, since these are two of the variable parameters to be tested. Furthermore, varying the number of cameras makes it possible to check for the accuracy difference between two, three and four cameras. All of the input values to the script are provided in table 3.3.

The analysis of angles between the cameras are done for a set of two, three and four cameras. The choice of number of cameras is done due to Maas [12] accuracy estimation implying more than four cameras will not improve the accuracy. Five different numerical analyses are being performed: A study with 1) two cameras, 2) three cameras located asymmetrically, 3) Three cameras located symmetrically in a triangle, 4) Four cameras located asymmetrically, and 5) Four cameras located asymmetrically in a square. All of the five different camera setups are shown in figures 3.4 - 3.6, including corresponding orientation of angles to be varied. A model of the point to track is represented by the black dot in all the illustrations, which the cameras are

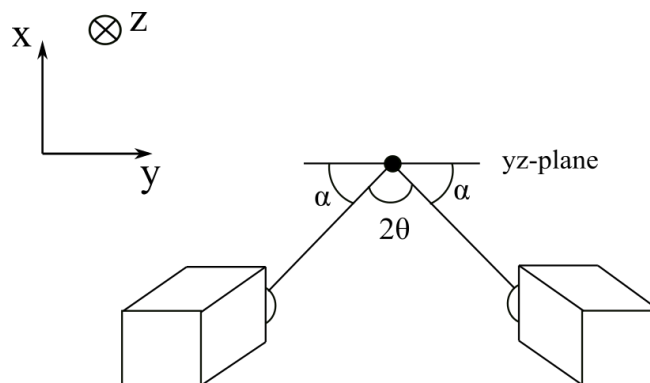


Figure 3.4: Camera position of two cameras

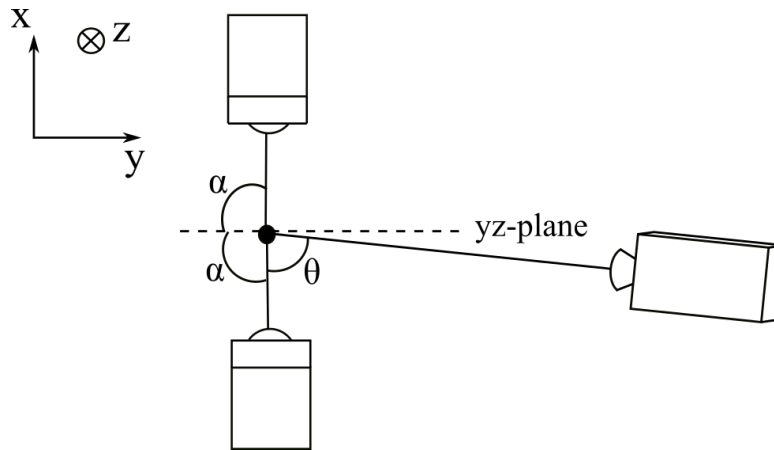


Figure 3.5: Asymmetric camera position of three cameras

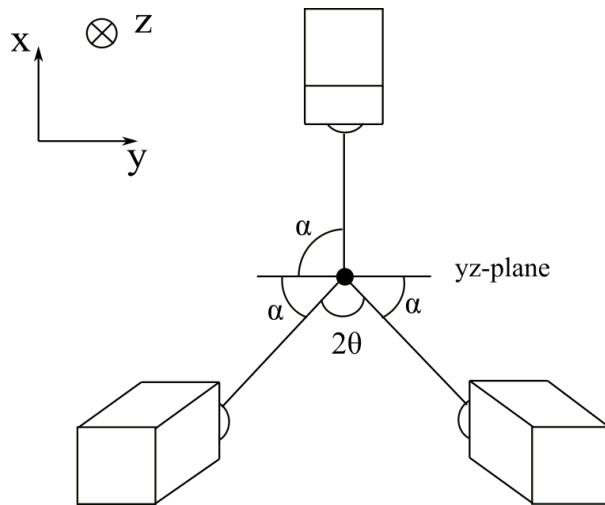


Figure 3.6: Symmetric camera position of three cameras

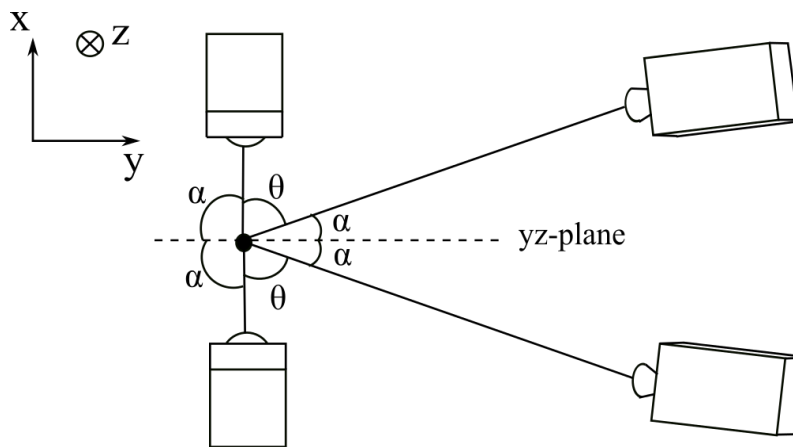


Figure 3.7: Asymmetric camera position of four cameras

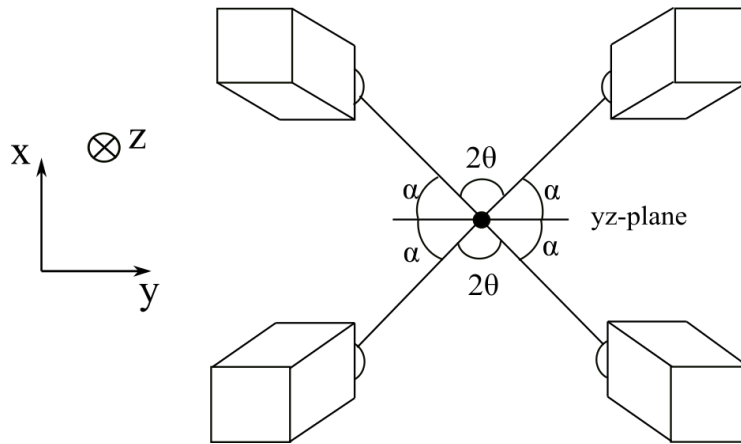


Figure 3.8: Symmetric camera position of four cameras

oriented about.

For each camera setup, a Monte Carlo simulation is performed with 1000 iterations. This implies 1000 iterations through all θ and α variations. As reported in section 2.3, in a perfect calibration the lines of sight from the calibration points will intersect. The camera parameters of this analysis are fictive and the lines from the cameras will intersect perfectly. However, this is never the case in real calibrations, and to account for noise and other sources of errors, a set of random values are applied to the calibration points. These errors have a value between -1 and 1 pixel, and are applied as an addition of $2\text{rand}(1) - 1$ to the x and y coordinate of the image pixels, before lines of sight are calculated. `rand()` is an integrated function in MATLAB returning a random value between 0 and 1. Furthermore, calculations of lines of sight and following intersection point for each calibration point, i.e. all detected corners of the checkerboard in the images in pixel values, are performed. Post processing of results are conducted with MATLAB and provided in chapter 4.

3.3 Point Tracking

After the calibration parameters are obtained from the approach described in section 3.1, it is possible to perform a tracking analysis of a moving object. This involves capturing a series of images of the object marked with a point. Each pixel position of the point centers can be reconstructed into world coordinates through the triangulation procedure explained in section 2.3. Performing this, the point must be detected in each image in order to obtain its center

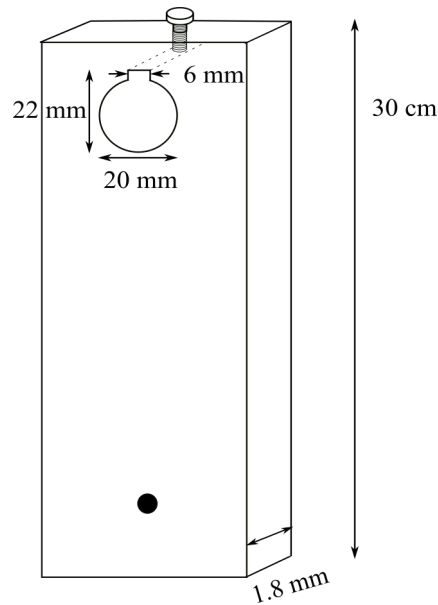


Figure 3.9: Sketch of plank used for tracking experiment. Dimensions are given on sketch.

coordinates. Different methods can be applied to do so, but circle detection with MATLAB is the approach used in this thesis.

How the parameters of the function are set, vary from image sequence to image sequence. Especially radius span and sensitivity must often be regulated in terms of how the object is angled. A point viewed from an angle can for instance look like an ellipse or a skewed circle.

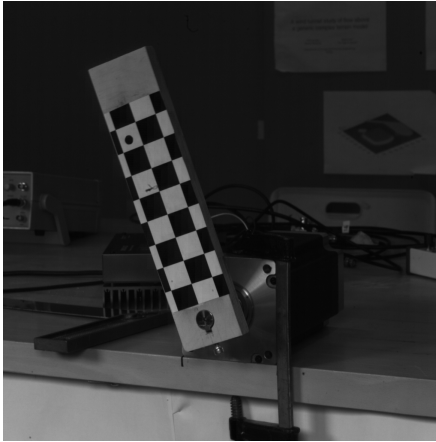
3.3.1 Experimental point tracking

Two Photron Fastcam cameras are mounted on a 150 cm long traverse and angled towards the tracking object. The camera lenses are 60 mm Nikon lenses, and the tracking object is a wooden plank of length 30 cm depicted in figure 3.9. The plank is connected to a stepper motor. Power to the stepper motor is provided from a signal generator through a stepper motor controller. Setup of the plank and the stepper motor is shown in figure 3.11. A black circular point is glued onto the plank as depicted, which is the point to be tracked. Two different points of different radii are used to evaluate the importance of size of circular mark. The points have diameters of 9.5 mm and 5.5 mm, respectively.

Primarily, the cameras are calibrated as described in the first section of this chapter. The different object setups are provided in table 3.4. Multiple sequences of the plank rotation are

Table 3.4: Different positions of cameras and plank for point tracking experiment

TRACKING	DEPTH	CAMERA DISTANCE	CAMERA ANGLE	OBJECT ANGLE	D_{point}
Arc 1	130 cm	120 cm	25°	-	9.5 mm
Arc 2	130 cm	120 cm	25°	-	5.5 mm
Arc 2	130 cm	86 cm	18°	-	5.5 mm
Circle 1	160 cm	125 cm	20°	20°	5.5 mm
Circle 2	160 cm	125 cm	20°	35°	5.5 mm
Circle 3	160 cm	125 cm	20°	45°	5.5 mm



(a) Arc tracking setup



(b) Full rotation tracking setup

Figure 3.10: Setups of tracking experiments

captured; only an arc of the circular path, and the whole circular path from an angled point of view. The two different points of view are depicted in figure 3.10. In addition, capturing sequences of the rotating plank from different camera angles, is done to examine how the depth of the circular path can be determined.

The distance between the camera traverse and the plank are 120 cm for the arc detection, and 160 cm for the full circular sequence. Limitations in the focal lens determine at what distance the whole plank (about 60 cm diameter) can be captured in one image frame. Also the distance between the cameras are varied in order to obtain lines of sight from various angles.

All the sequences of tracking the arc, are rotated on a frequency of approximately 1500 Hz, while the whole circle sequences are rotated at a frequency of about 2000 Hz. The stepper motor has 200 steps per revolution, while the controller has several micro stepping modes [25], deciding how many pulses each step are divided into. For all tracking sequences, the micro step mode is set to $\frac{1}{128}$. This implies a rotational speed of 12 to 18 seconds per revolution.

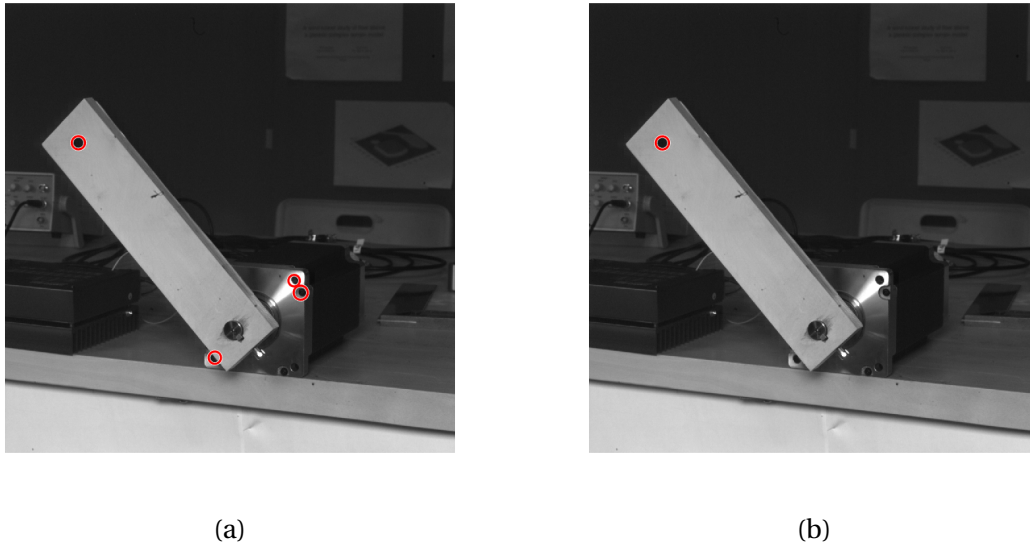


Figure 3.11: Too many circles(a) and correct circle(b) detected in image from tracking sequence

The stepper motor is securely fastened to a table. However, the table is not fastened to the floor properly, causing the stepper motors movements to vibrate the table. These movements are also causing the plank to vibrate, and errors will be expected to occur in the back projection of the path. Arc 2 is therefore a sequence of only 94 frames, 47 per camera, where the stepper motor is off, and the plank is moved by hand one step at the time. One frame is captured by both cameras of each step position, to be able to estimate the error occurring due to inner stepper motor vibrations. The remaining sequences of an arc tracking consists of 300 frames per camera, and for the full rotation tracking, between 800 and 900 frames per camera. The cameras captures images through the computer program Photron FASTCAM Viewer at a speed of 50 frames per second at a image resolution 1024x1024 pixels.

After the tracking sequence, the images are loaded into a MATLAB script to detect the circular mark with `imfindcircles()`. To avoid detection of wrong circles, as depicted in figure 3.11, a constraint is given in the algorithm. This constraint is a start point located within a radius of 10 pixels from the wanted circle. If a detected circle are further away than 10 pixels from the start point, it will be ignored. The detected circle then becomes the new start point. The circles detected in all images are saved, and reconstructed into world coordinates by finding lines of sight and intersection point. Tracking results are then plotted and investigated with MATLAB.

Chapter 4

Results and Discussion

The results from the experiments and numerical analysis performed are presented in this chapter. The presentation of the studies follow the same order as arranged in the previous chapter: Primarily the experimental camera calibration results; the numerical angle analysis; and last, the experimental tracking results. A discussion of the results follows after each subsection. Both graphical and tabulated findings are provided accomplished by the most important information necessary to conclude from the analyses.

4.1 Experimental camera calibration

The experimental camera calibration was performed in order to obtain the expected accuracy of reconstructing world points from 2D image points through back projection. The aim was to determine the error connected to the deviation of the reconstructed 3D points from the actual world points. Furthermore, the camera parameters obtained in the calibration experiment, was

Table 4.1: Table of camera calibration results

APPROACH	DIR	MEAN [mm]	STD. DEVIATION [mm]	VARIANCE [mm ²]
APPROACH (A)	x	0.05	0.07	0.006
	y	0.04	0.05	0.003
	z	0.08	0.11	0.012
APPROACH (B)	x	0.05	0.07	0.004
	y	0.04	0.05	0.003
	z	0.10	0.15	0.023

used in the numerical analysis presented in next section.

Recalling from section 3.1.1 only one camera was used to simulate a three camera setup. The image sequences necessary for the calibration were captured by two different approaches to determine any significance of difference. The two image capturing methods were

- (A) one calibration object orientation was captured from all three camera positions, before the object position was changed
- (B) all 15 object orientations were captured form one camera position, before the camera was moved and procedure repeated

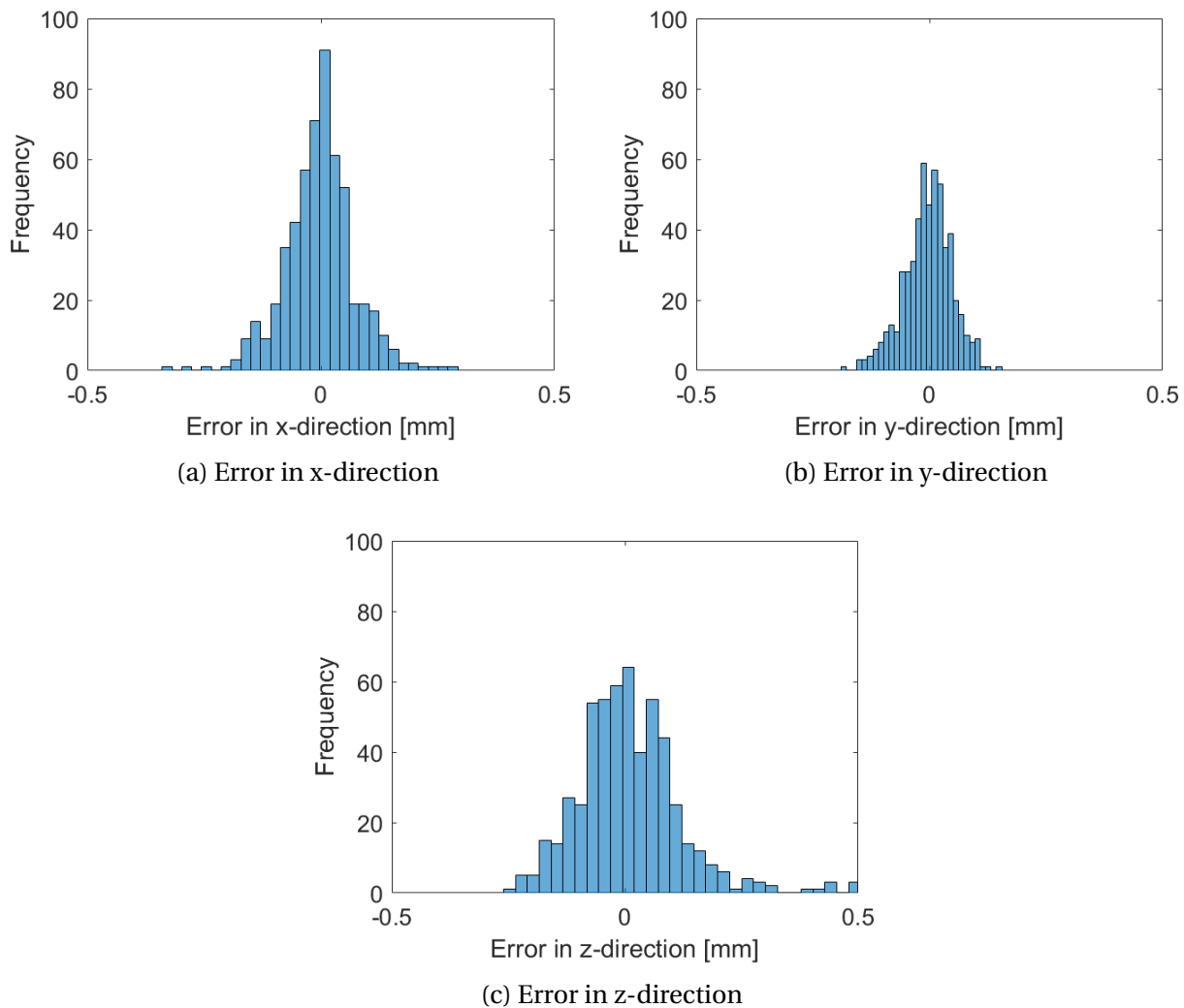


Figure 4.1: Approach (A). Histograms of errors in x, y and z-direction obtained from capturing one object orientation from all camera positions at a time

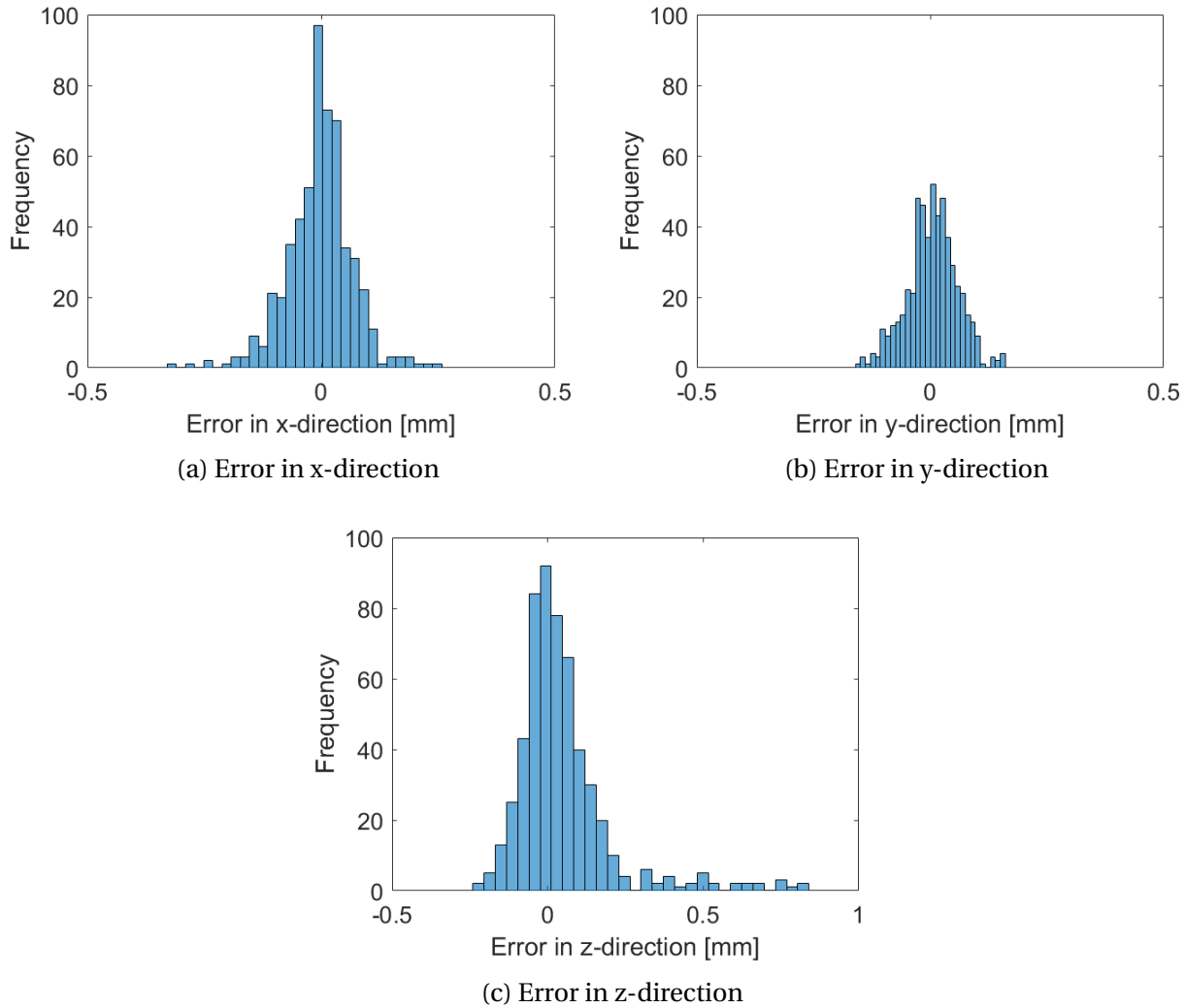


Figure 4.2: Approach (B). Histograms of errors in x, y and z-direction from the procedure where all object orientations are captured from one camera position at a time

The code provided in appendix section C.1, recognises the checkerboard corners as calibration points. However, in two of the images captured with approach (B) the corners could not be detected. Therefore, these images needed to be removed from the calibration of sequence in both approaches to make them comparable. A total number of 13 images were therefore used in the two sequences.

Results of the reconstruction errors are provided in both table 4.1 and as histograms in figures 4.1 and 4.2, for approach (A) and (B), respectively. The mean values in the table are the mean of the absolute value of the errors, implying the expected error in mm away from the actual point. The histograms show the distribution of the reconstruction error around the ac-

tual world position in x , y and z -directions. There are 30 pillars in each plot referring to the quantities of errors of that specific magnitude. From 42 calibration points in each image, this corresponds to a number of $42 \times 13 = 546$ back projections and also data points in the plot.

Reconstruction errors of approach (A) are provided in figure 4.1. Error in x -direction have an absolute mean of 0.05 mm and standard deviation of 0.07 mm. The distribution spreads out to about ± 0.3 mm in extremal values, but these values are of low frequency. In y -direction, the reconstruction error are lowest, implying the most accurate back projection direction. Mean reconstruction error are 0.04 mm and standard deviation 0.05 mm. Results in z -direction are least accurate, and have extremal values up to 0.5 mm away from actual world point. The absolute mean and standard deviation are 0.08 mm and 0.11 mm, twice as high error as in y -direction.

In figure 4.2, histograms of sequence (B) are depicted. All three figures are visually showing a high frequency of errors very close to zero. Results in x and y -direction are equal as in sequence (A), however, in z -direction, the error is 0.02 mm larger.

From table 4.1, the differences in results between (A) and (B) are negligibly small in both mean values, standard deviation.

Contour plots of total reconstruction errors corresponding to each image used in the camera calibrations are provided in figure 4.3. The total error is defined as $E_{tot} = \sqrt{E_x^2 + E_y^2 + E_z^2}$. Image three are providing significantly larger errors than the rest of the images, especially in approach

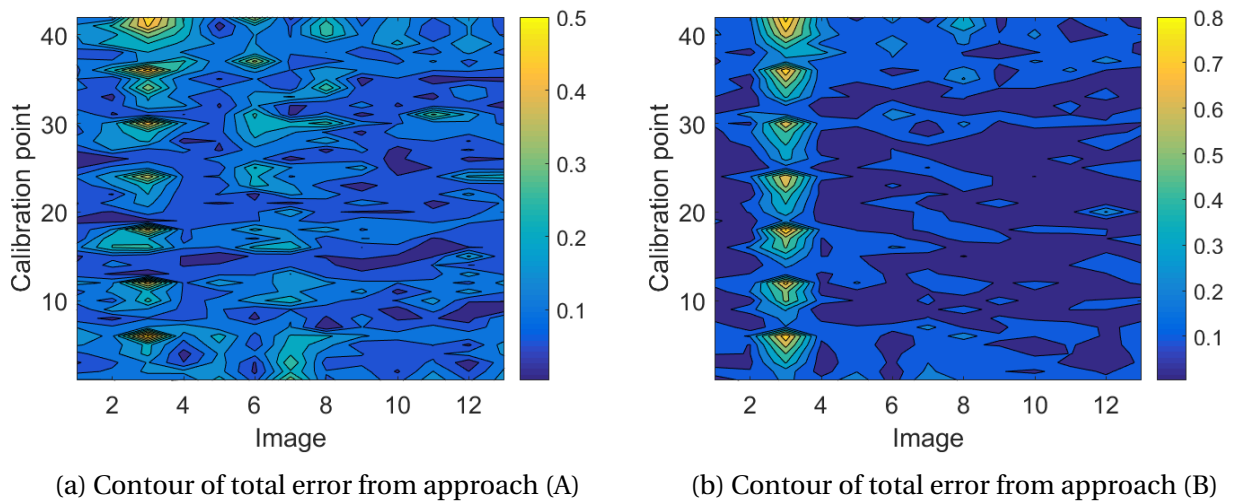


Figure 4.3: Contour plots of total reconstruction errors from images used in the camera calibration

(B). The calibration points corresponding to these are 6, 12, 18, 24, 30, 36 and 42, and are the points corresponding to the right edge of the checkerboard. The checkerboard was then angled 30° to the right.

4.1.1 Discussion camera calibration

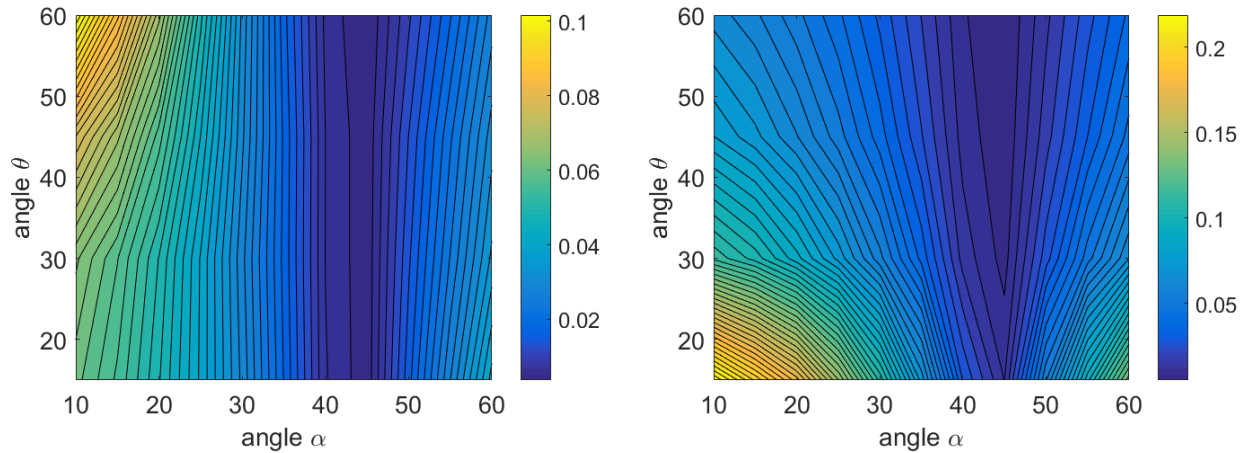
Image capturing method (A) and (B) does not differ significantly in order to result in one method being better than the other. The main difference lies in estimating the depth, being 0.02 mm in difference of mean. This difference can be a result of image number 3, seen in figure 4.3b, having much higher reconstruction errors than the rest of the images. Despite the standard deviation difference of 0.05 mm, the errors are too small to result in large deviations in accuracy of the two methods. From this experiment, one can expect reconstruction errors due to calibration of magnitude 0.05 mm in x -direction, 0.04 mm in y -direction, and up to 0.1 mm in z -direction. However, the standard deviations are somewhat close to the actual mean values, implying even larger errors can occur. Again, the depth reconstruction are particularly exposed for large variations in results with standard deviations of magnitude 0.1 mm.

Images showing larger reconstruction errors in contour plots can be omitted from the calibration, and may result in better accuracy. It showed that calibration objects oriented at a sharp angle did not enhance the reconstruction error of the calibration. Adding more object orientations at different elevations, might decrease errors. Especially moving the checkerboard to cover the edges of the frame, can result in better accuracy.

4.2 Numerical angle analysis

The numerical analysis was performed to determine the number of cameras in a setup and the corresponding angle between them, resulting in most accurate back projection. Results from this analysis was meant to be applied in the water tank experiment, but are for now only preparation work to establish expected errors related to camera angles.

Contour plots of mean error corresponding to the different angles between the variety of camera setups are depicted in figures 4.4 to 4.8. Only plots for error in y -direction and z -direction are provided. The depth estimation is the most relevant for the tracking procedure,



(a) Contour of error in y -direction, color bar values are in mm (b) Contour of error in z -direction, color bar values are in mm

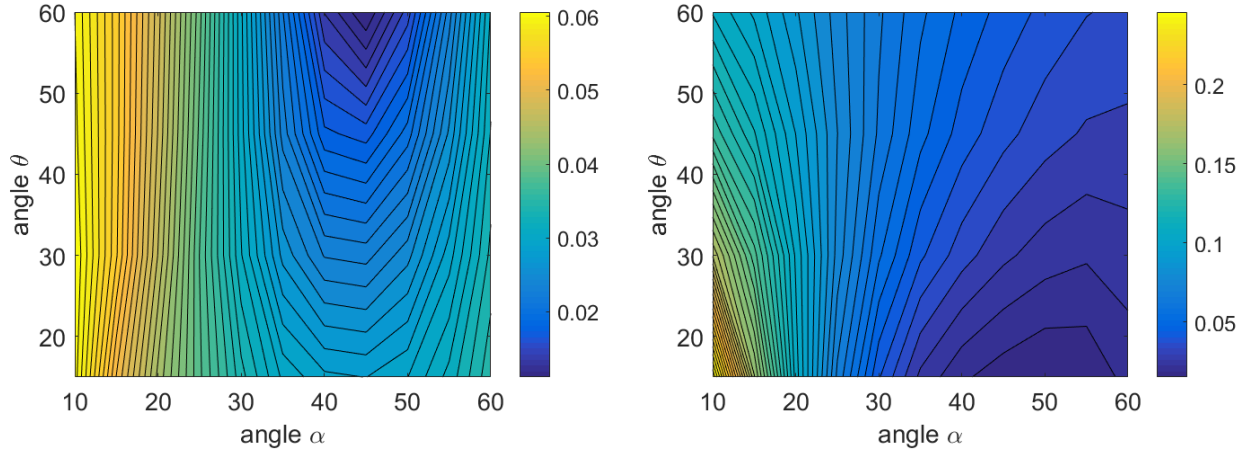
Figure 4.4: Mean error contour for corresponding angles α and θ for a two-camera setup

due it being difficult to achieve perfectly. The errors are smallest in x -direction from table 4.2, and therefore of least importance of total reconstruction error. Hence, contour plots in x -direction are not provided.

In the two-camera setup provided in figure 4.4, the lowest error values are seen for α angles of about 45° . Looking solely in y -direction, the low values of purple color, are independently of variations in θ , and therefore only dependent of camera setup in vertical direction. The highest however, are present for large θ angles, and small α 's, respectively. In the z -direction, low θ and α results in largest reconstruction error. Best results are also present for higher θ for the depth estimation.

For the asymmetric three-camera setup showed in figure 4.5, the results are more dependent of θ than in the two-camera setup. Results in y -direction shows smallest reconstruction errors of θ close to 60° and $\alpha = 45^\circ$. In z -direction, best results are obtained for an α between 40° and 60° , over a span of $15^\circ < \theta < 30^\circ$. This implies a large variety in camera orientations corresponding to low errors.

The symmetric three-camera setup are independent of θ in both y and z -direction. Similarly, the magnitude of the maximum errors are smaller than for the previous setups, implying an enhanced depth estimation. The smallest error in y -direction occurs at an α of 45° , while for $40^\circ < \alpha < 50^\circ$ in z -direction. This error width of α does not change with θ in y -direction, but



(a) Contour of error in y-direction, color bar values are in mm (b) Contour of error in z-direction, color bar values are in mm

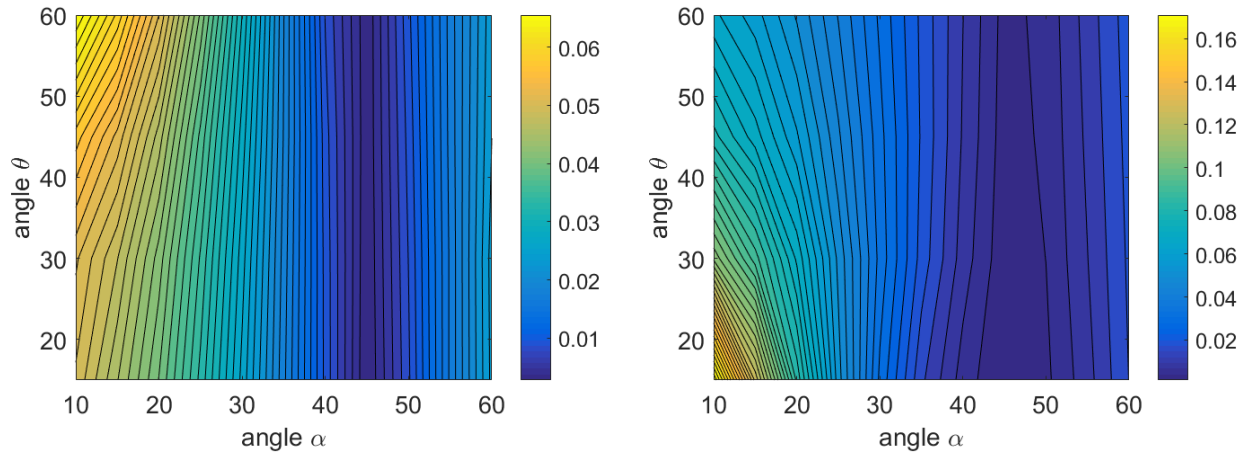
Figure 4.5: Mean error contour for corresponding angles α and θ for an asymmetric three-camera setup, see figure 3.5

becomes more narrow for bigger θ angles in z -direction. At an angle of $\theta = 60^\circ$, the optimal α is 45° . On the contrary, small α angles leads to largest error in both y and z -direction.

In the setups of four cameras, both the asymmetric and symmetric results, depicted in figures 3.7 and 3.8, are showing similar trends as the symmetric three camera setup seen in figure 3.6. The smallest error in both y and z -direction corresponds to an α of about 45° , more or less

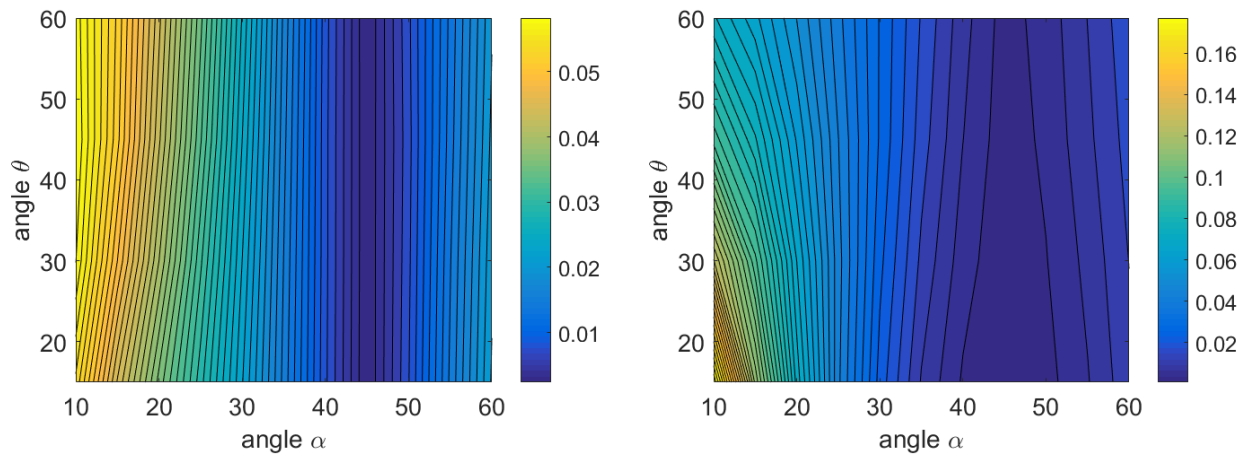
Table 4.2: Minimum and maximum mean error values

CAMERA SETUP	DIR	MIN. ERROR [mm]	MAX. ERROR [mm]	STD. DEVIATION [mm]
2 CAMERAS	x	0.003	0.252	0.034
	y	0.004	0.103	0.048
	z	0.005	0.229	0.043
3 CAMERAS ASYMMETRIC	x	0.039	0.065	0.057
	y	0.011	0.062	0.039
	z	0.015	0.250	0.077
3 CAMERAS SYMMETRIC	x	$5.6e-4$	0.047	0.023
	y	0.003	0.067	0.037
	z	0.001	0.174	0.037
4 CAMERAS ASYMMETRIC	x	$3.6e-4$	0.044	0.021
	y	0.002	0.060	0.032
	z	$7.6e-4$	0.181	0.037
4 CAMERAS SYMMETRIC	x	$3.7e-4$	0.041	0.020
	y	0.003	0.076	0.036
	z	0.001	0.130	0.029



(a) Contour of error in y-direction, color bar values are in mm (b) Contour of error in z-direction, color bar values are in mm

Figure 4.6: Mean error contour for corresponding angles α and θ for a symmetric three-camera setup, see figure 3.6

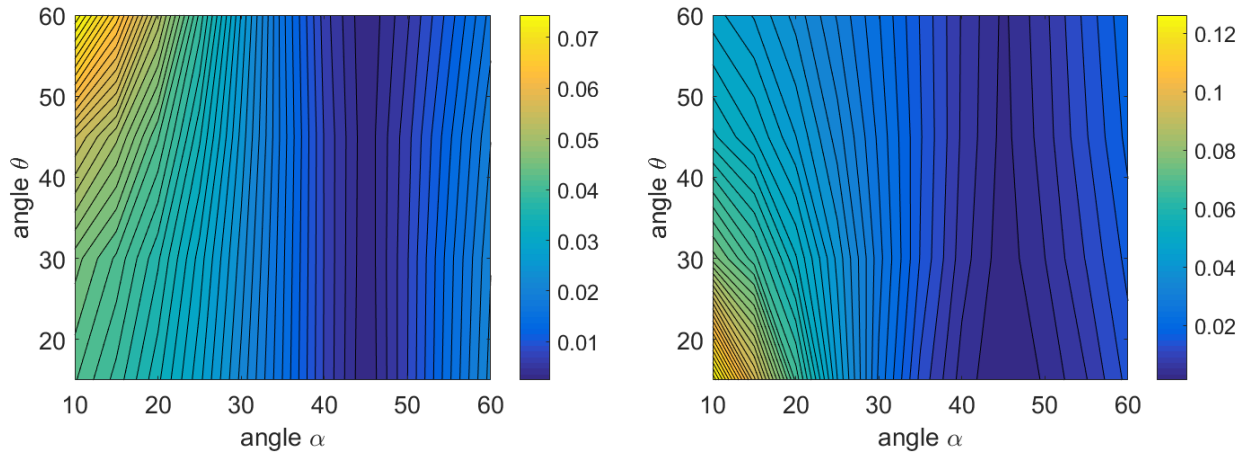


(a) Contour of error in y-direction, color bar values are in mm (b) Contour of error in z-direction, color bar values are in mm

Figure 4.7: Mean error contour for corresponding angles α and θ for an asymmetric four-camera setup, see figure 3.7

independent of θ . Furthermore, the largest inaccuracies of the symmetric setup can be found at the smallest and largest θ angles of 15° and 60° in y and z -directions, respectively. These occur at an angle of $\alpha = 10^\circ$. Large errors of the asymmetric four-camera setup are appearing equally as for the symmetric three-camera setup provided above.

All the minimum and maximum values are provided in table 4.2. A decreasing trend of maximum values can be seen from top to bottom of the table, where the two-camera setup having



(a) Contour of error in y-direction, color bar values are in mm (b) Contour of error in z-direction, color bar values are in mm

Figure 4.8: Mean error contour for corresponding angles α and θ for a symmetric four-camera setup, see figure 3.8

highest error and the symmetric four-cameras having smallest. Neither of the minimum values are of significant magnitude ($< 10^{-3}$ mm).

Figures 4.9 to 4.11 are showing plots of the total root mean squared (rms) error versus angle α . The total rms error is here defined as $error_{rms} = \frac{\sqrt{\sum x_i^2 + y_i^2 + z_i^2}}{N}$, where N is the number of intersection points. Figures 4.9, 4.10b and 4.11 are all showing minimum point at α equal 45° .

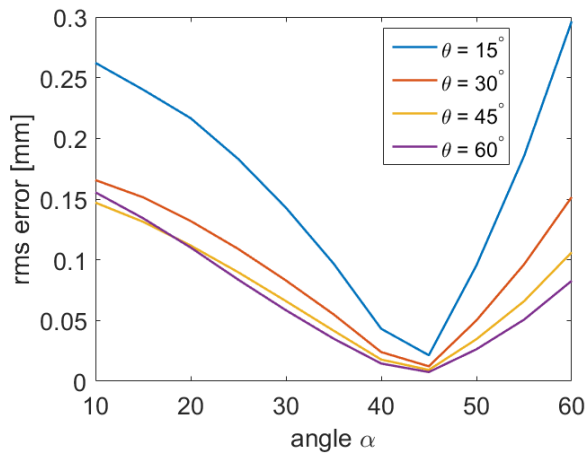


Figure 4.9: Total rms error of a 2 camera setup

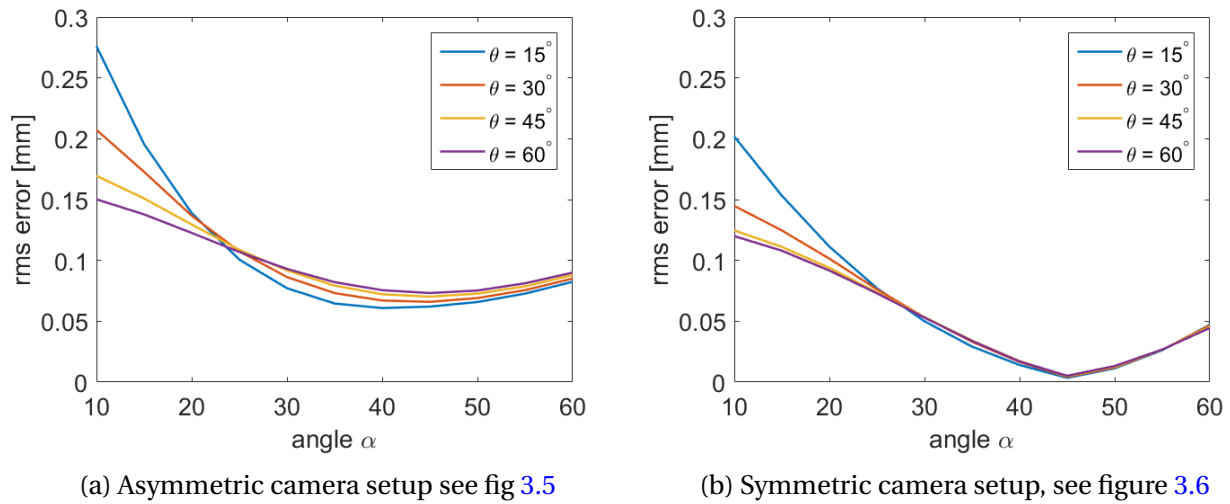


Figure 4.10: Total rms error for 3 cameras in; (a) asymmetric setup; and (b) symmetric setup for corresponding α and θ angle.

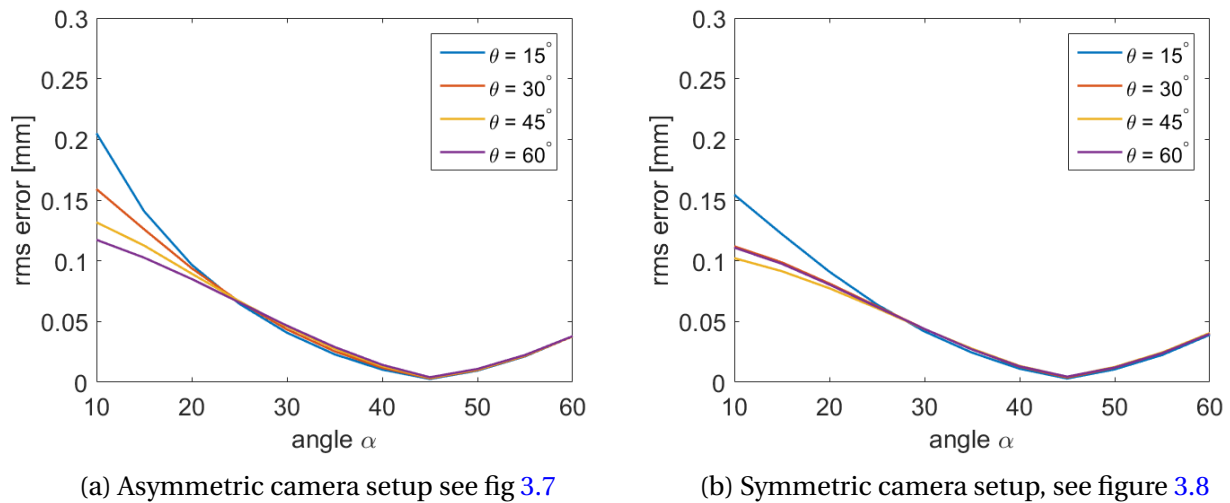


Figure 4.11: Total rms error for 4 cameras in; (a) asymmetric setup; and (b) symmetric setup for corresponding α and θ angles

4.2.1 Discussion numerical analysis

Recalling the conclusion of Lawson et. al [11] for a stereo camera setup having the smallest error at an angle α of 45° , it is not surprisingly to obtain similar findings of an numerical analysis of a two camera setup. Furthermore, it is interesting how the rest of the remaining four camera setups also obtain the most accurate back projection at an angle α of 45° , completely independent of θ for most setups. Obviously, an orthogonal view of the cameras in the vertical direction

leads to the most accurate results. Additionally, the results of the total reconstruction error of symmetric three-cameras, and both of the four-camera setups showed in figures 4.10b to 4.11b, are seemingly equal. Obtaining similar results as Maas [12] can therefore be concluded.

Interestingly is the large error obtained for an asymmetric three camera setup. From figure 4.10a, the minimum value at $\theta = 15^\circ$ does not reach an error below 0.05 mm, while the other results obtain almost zero reconstruction error at their minimum point. Not applying this kind of camera setup of a tracking procedure is therefore of preference in future calibration experiments.

4.3 Point tracking experiment

The point tracking experiment was performed to establish how well the path of a rotating plank could be reconstructed. That included reconstruction of the path radius and depth, and to check for vibrations of the plank during rotation. Experimental results from reconstructing the path of the plank are provided in this section. The results reflect tracking of a point glued to the plank of diameter 9.5 mm and 5.5 mm as described in section 3.3. Primarily, the results from tracking of an arc of the rotation are presented. Secondly, results from the whole rotational sequences are provided. Important results from both procedures are the reconstructed radius of the path, and how well it corresponds with the actual radius, and depth estimation.

4.3.1 Arc of rotation

Three different arc tracking sequences were performed. Within each sequence, two comparable series with equal radius, point size and frequency were conducted. In all sequences, the distance from the camera traverse to the tracking object was 130 cm. The plank was rotated in the xy -plane approximately parallel to the camera traverse, meaning the rotational plane not having significant depth difference in z -direction. Recalling table 3.4 in section 3.3, the main parameters separating the three experiments were: the camera locations; and the size of the traceable point. In table 4.3, the results from the three different sequences are presented with values of mean radius and depth predicted.

The real radius R_{real} is the measured distance between the center of the rotational axis of

the plank and the center of the tracking point. The estimated radius $R_{\text{estimated}}$ however, is the calculated radius of a circle fitted to the tracked path, i.e. experimental results. Fitting a circle to the arc is obtained through a multi-step procedure. This procedure is used for both the arc reconstruction, and reconstruction of the full rotation presented in the next section. The steps are the following:

- A plane is first fitted to the reconstructed center points through a least squares fit.
- If the fitted plane is not located accurately in the xy -plane, a rotation is applied to the center points.
- The rotation matrix is obtained through the calculation provided in Appendix section ??, using normal vectors.
- After rotating the fitted plane and center points onto the xy -plane, a 2D circle can be fitted to the "new" center points. The code used for circle fit was written by Bucher [2].

Plots of the reconstructed paths of the parallel xy -plane are depicted in figures 4.12, 4.14 and 4.16. Each figure displays two plots being results from a sequence with equal actual radius, equal point size and same camera distance. Figure (a) corresponds to the first row of data in each arc sequence in table 4.3, and similarly (b) corresponds to the second row.

Figures 4.13, 4.15 and 4.17 shows the fitted plane rotated onto the xy -plane. The corresponding rotated points are plotted as the fluctuating graph on the plane. These fluctuating values would follow the plane accurately if the path reconstruction was perfect, and the rotating plank was non vibrating. However, this is not the case, and the variations of depth is here illustrated as errors. The mean values of the errors and the standard deviations are provided in table 4.3. Each plot corresponds to the same series, (a) and (b), as described for the path plots above.

Table 4.3: Tracking results from reconstruction of arc

SEQUENCE	NO	R_{real}	$R_{\text{estimated}}$	STD. R	MEAN DEPTH	STD. DEPTH
Arc 1	(a)	195 mm	195.7 mm	± 0.093 mm	0.12 mm	± 0.15 mm
	(b)	195 mm	196.1 mm	± 0.141 mm	0.22 mm	± 0.32 mm
Arc 2	(a)	195 mm	194.0 mm	± 0.107 mm	0.13 mm	± 0.10 mm
	(b)	195 mm	193.1 mm	± 0.179 mm	0.11 mm	± 0.09 mm
Arc 3	(a)	195 mm	195.3 mm	± 0.070 mm	0.18 mm	± 0.15 mm
	(b)	195 mm	193.8 mm	± 0.075 mm	0.16 mm	± 0.13 mm

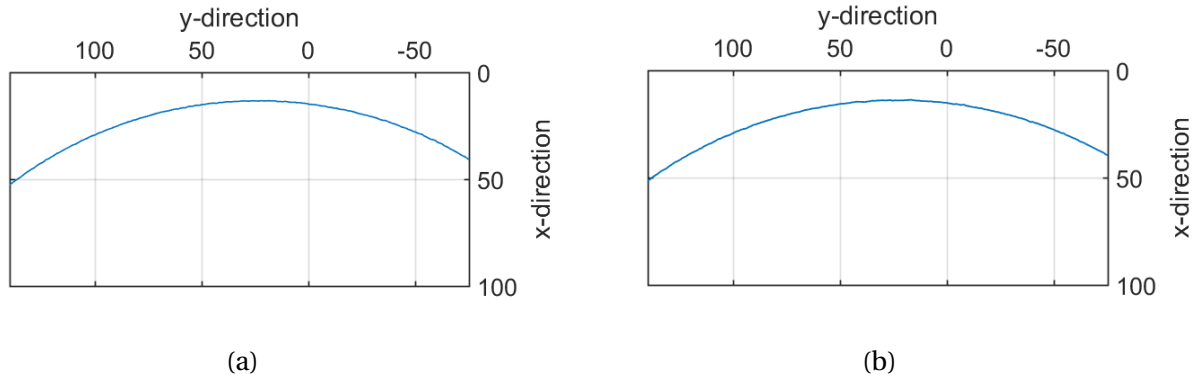


Figure 4.12: Projected paths of sequences of Arc 1

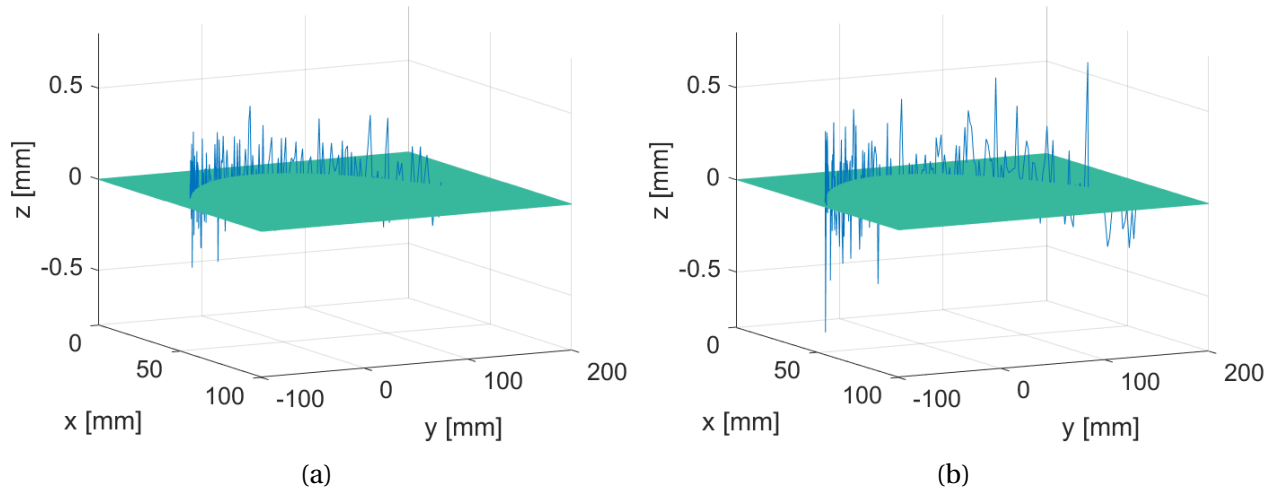


Figure 4.13: Fluctuating depth values of reconstructed points in sequences of Arc 1

Arc 1

Starting with the results from sequence Arc 1. The distance between the cameras was 120 cm, and the distance from the camera traverse to the plank was 130 cm. The calibration estimated a total angle of 50° between the cameras, corresponding to a θ of 25° compared with the numerical analysis in section 4.2. The diameter of the tracking point was 9.5 mm, the largest point used in all sequences.

The reconstructed paths plotted in figure 4.12 are visually showing very similar results. Sequence Arc 1 (a) are showing best results in both estimating the radius, and the depth as seen in table 4.3. The radius estimation of (a) being 195.7 mm is only 0.7 mm larger than the actual radius. Sequence (b) estimated the radius to be 196.1, which is 1.1 mm larger than the actual

radius.

Comparing figures 4.13a and 4.13b, the depth error of (a) is slightly smaller than the error estimated in (b). This can be seen of the plots from the large fluctuations, and is also provided in table 4.3 where the difference between the mean values are 0.1 mm.

Arc 2

The two sequences of Arc 2 were captured in two different ways, in order to find out how the inner vibration affects the accuracy of the depth estimation. In sequence (a), the stepper motor was run by the controller, and a number of 300 images was captured by each camera. In sequence (b) however, the stepper motor was not started, but physically moved one step at a time between each frame capturing. This resulted in a total of only 47 frames per camera, and no vibrations caused by the motor.

The results from sequence Arc 2 are provided in figure 4.14 and 4.15. As seen from the projected paths in figure 4.14, the two plots corresponds very good. Sequence (a) predicts the radius slightly better than sequence (b). The two radii are 194 mm and 193.1 mm, being different from actual radius by 1 mm and 1.9 mm, respectively. In figures 4.15, the depth fluctuations are provided. It is clear from figure 4.15b that less images are used in the reconstruction of the path, and the extremal values looks smaller than in figure 4.15a. The depth fluctuations have a mean of 0.11 mm and standard deviation of ± 0.09 mm. Sequence (b) predicts a mean depth error of 0.13 mm with a standard deviation of ± 0.10 mm, and are only 0.02 mm larger than (b). These

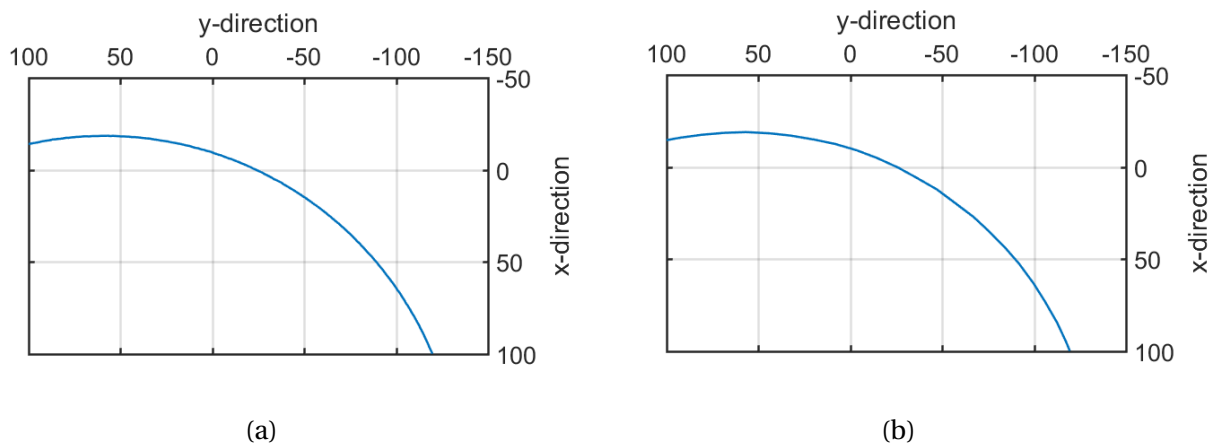


Figure 4.14: Projected paths of sequences of Arc 2

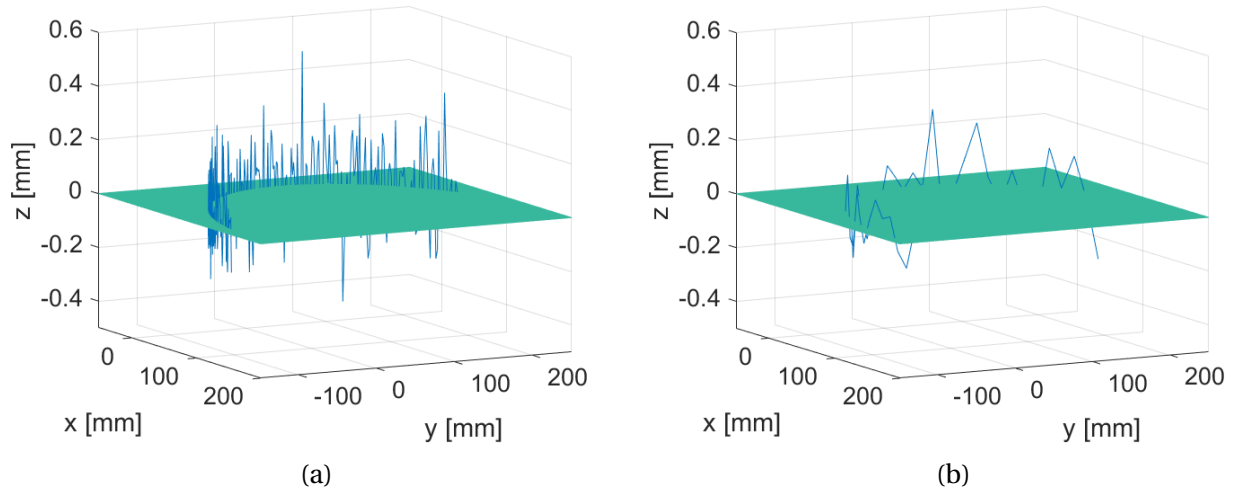


Figure 4.15: Fluctuating depth values of reconstructed points in sequences of Arc 2

values are the smallest depth errors obtained from all three arc tracking experiments.

Arc 3

The results from the last arc tracking series, Arc 3, are depicted in figure 4.16 and 4.17. Arc 3 differ from Arc 1 and 2 by having a smaller angle between the cameras, meaning a shorter distance between the cameras on the traverse. The angle and distance between the cameras were 18° and 86 cm, respectively. Moreover, the diameter of the tracking point was 5.5 mm.

From table 4.3 the radius estimations are 195.3 mm with a standard deviation of ± 0.07 mm from sequence (a) and 193.8 mm and standard deviation of ± 0.075 mm from sequence (b). The

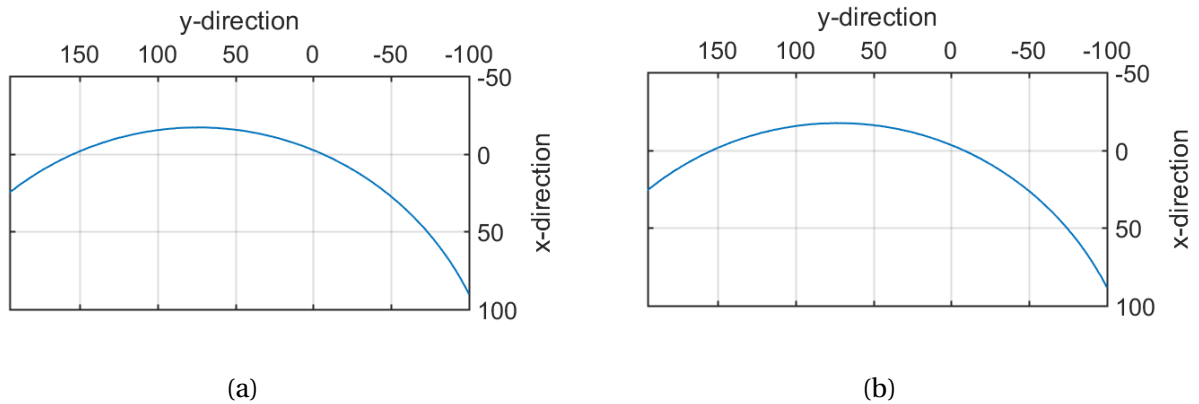


Figure 4.16: Projected paths of sequences of Arc 3

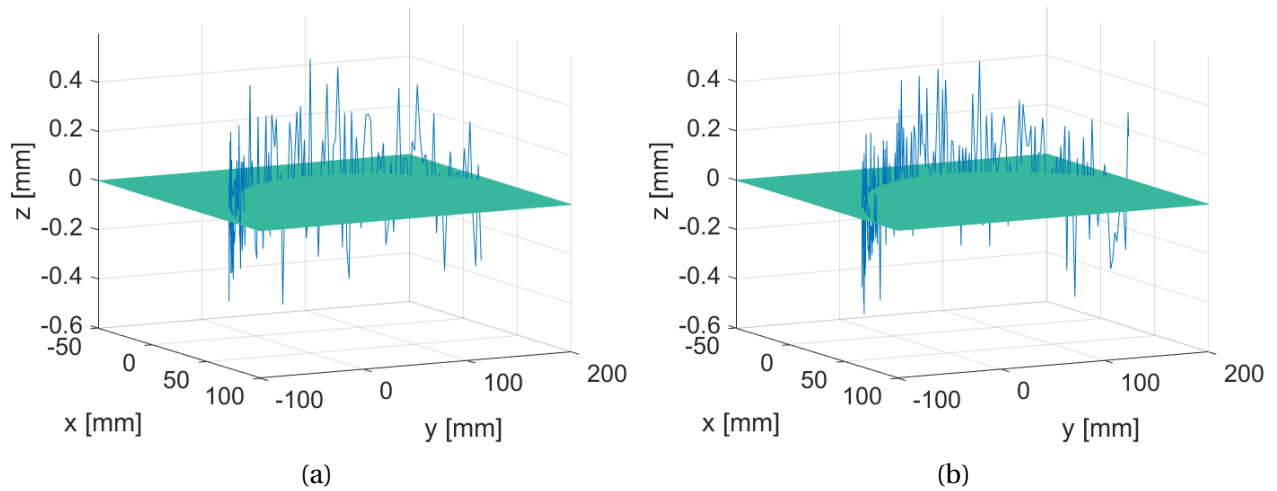


Figure 4.17: Fluctuating depth values of reconstructed points in sequences of Arc 3

two plots of the reconstructed path can be seen in figures 4.16a and 4.16b. The depth estimations are quite similar in both sequences, and can be seen in figures 4.17a and 4.17b. The mean depth estimations are 0.18 mm and 0.16 mm, with standard deviations of ± 0.15 and ± 0.13 mm for (a) and (b) respectively.

4.3.2 Discussion of arc tracking

The aim of reconstructing the path of the arc has been reached. In the figures plotted of the arcs, its trajectories were well reconstructed visually and dimensionally. All of the sequences predicted the radius within 99.4% of the actual radius. The best prediction of radius was given from sequence Arc 3 (a) and was only 0.3 mm away from the actual radius. Recalling the calibration results from section 2.2, the expected error in x and y -direction was 0.05 mm and 0.04 mm, respectively, implying a small error is expected in both directions from the calibration alone.

Additional error can be resulted by the circle detection of the tracking point. If the center of the point is not determined properly, an error corresponding to pixel size is obtained. In these experiments the size of each pixel corresponds to 0.5×0.5 mm, and an error of similar size can be expected. Furthermore, measurements of the actual radii were done with a measuring tape. Inaccuracies of measuring the actual radius from the center of the point to the center of the rotating axis, are not excluded as a source of error. More exact measurements of actual radius should therefore be conducted in future experiments, to exclude this source of error completely.

More information on error analysis are provided in the last section of this chapter.

Depth estimation are showing most deviating results from actual plane depth. Samper et al. [22] concluded with largest reconstruction errors in z -direction, and suggested extra care taking when estimating the camera parameters in depth. No special care was taken of these parameters during calibration, but can be enhances in future work by calibrating the cameras using 3D objects. Results from the camera calibration of this thesis, provided in section 4.1 however, implies an expected error in z -direction of 0.1 mm. Accounting for an error of 0.1 mm due to calibration, the results of the tracking implies much better accuracy of depth estimation in all sequences.

Arc 2 was conducted to check if vibrations of the stepper motor caused higher errors in either x , y or z -directions. The extremal values depth reconstruction of Arc 2 (b) were seemingly lower than for the comparable sequence Arc 2 (a). However, the mean error value of 0.11 mm did not stand out from the other results. If this is trustworthy results or not can and should be tested in a future experiment. An easily implemented method to check for this accuracy, can be to track the plank from the side. More elaborated description of the method is provided in next chapter as future work.

Using a large tracking point contra a small tracking point did not results in better accuracy neither in radius estimation nor depth prediction. The results are seemingly equal, and using one point size rather than another, should therefore not change results significantly.

4.3.3 Full rotation tracking

In the full rotation tracking experiments the rotation axis of the plank is shifted an angle away from the parallel xy -plane, seen from figure 3.10b. This is done to get a variable depth of the circular trajectory to check how well the reconstruction of the depth can be done, and how this affects the radius estimation.

Table 4.4: Tracking results from reconstruction of circular path

SEQUENCE	ANGLE	R_{real}	$R_{\text{estimated}}$	STD. R	MEAN DEPTH	STD. DEPTH
Circle 1	20°	195 mm	193.3 mm	±0.44 mm	0.14 mm	±0.18 mm
Circle 2	35°	195 mm	193.3 mm	±0.35 mm	0.15 mm	±0.21 mm
Circle 3	45°	195 mm	192.9 mm	±1.41 mm	0.21 mm	±0.28 mm

Recalling table 3.4, the distance between the camera traverse and tracking object was increased from 130 cm to 160 cm. The rotation plane was shifted 20° , 35° and 45° clockwise from the earlier parallel path, and were divided into three sequences: Circle 1, Circle 2 and Circle 3, respectively. Unlike the arc tracking, only one series was performed per sequence. Results from radius and depth estimations from all sequences are provided in table 4.4.

The three sequences will be compared with each other to establish how the different angles affect the accuracy of the tracking. To estimate the radius of the circular path, the approach of rotating center points onto the xy -plane as described in the previous section, is applied. Plots of the circle fittings are provided in figures 4.18a, 4.18b and 4.18c. The blue circle reflects the

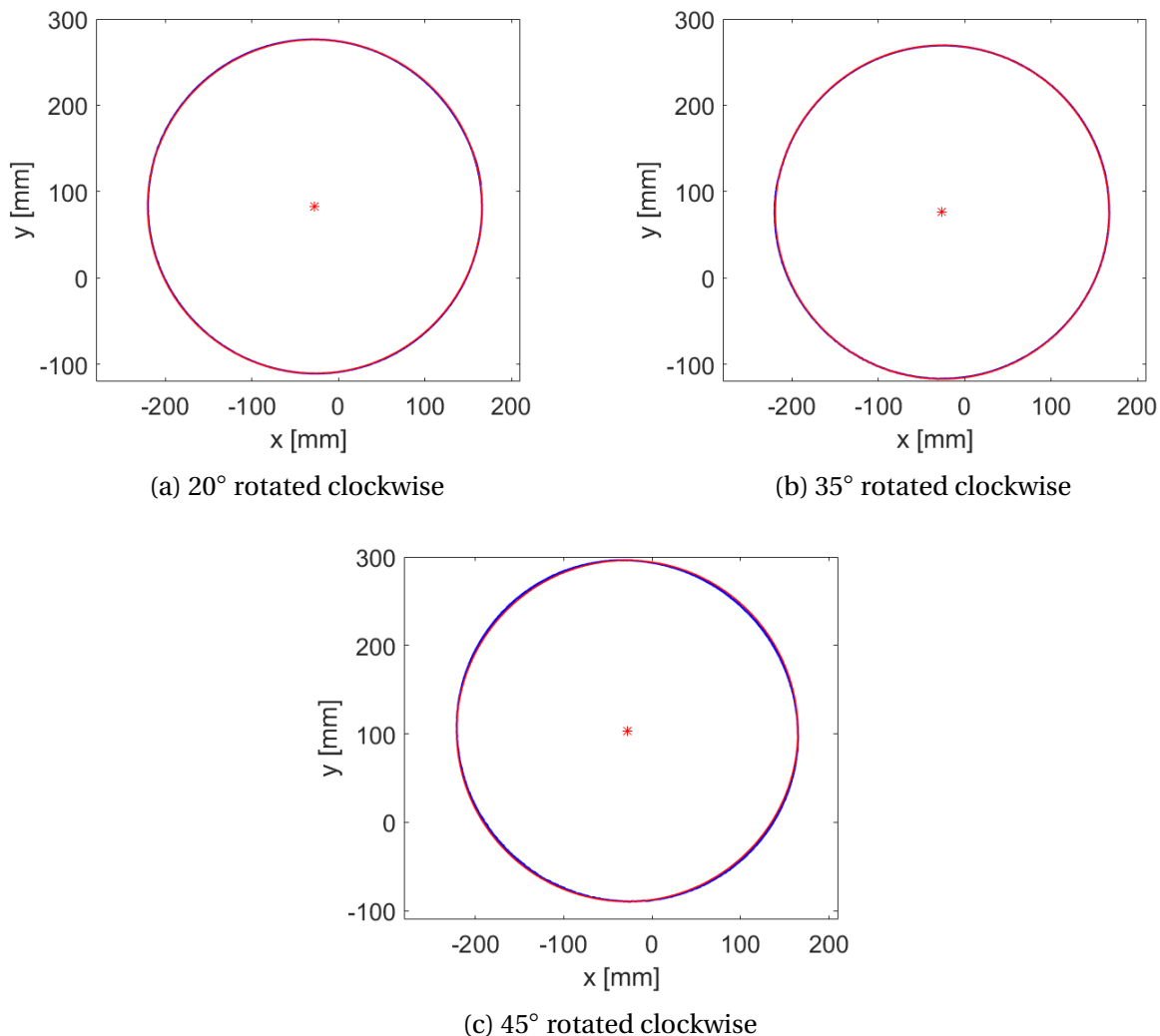


Figure 4.18: x and y coordinates of rotated tracking results in xy -plane.

tracked path, and the red circle is the fitted circle with radius equal to $R_{\text{estimated}}$ provided in table 4.4. The red star reflects the center point of the fitted circles.

Both results from Circle 1 and 2 matches the fitted circle seemingly well. This can be seen from no significant amount of blue color visible outside the fitted red circles. Looking at Circle 3 in figure 4.18c however, the blue path alternate of being inside and outside the red circle. This error can be associated with the large standard deviation of ± 1.41 mm of estimated radius. Nevertheless, comparing the mean error values, the difference between the three sequences are less than 0.2 mm. The radii estimation of Circle 1 and 2 are giving equal results of 193.3 mm, but have slightly different standard deviation of ± 0.44 and ± 0.35 mm, respectively.

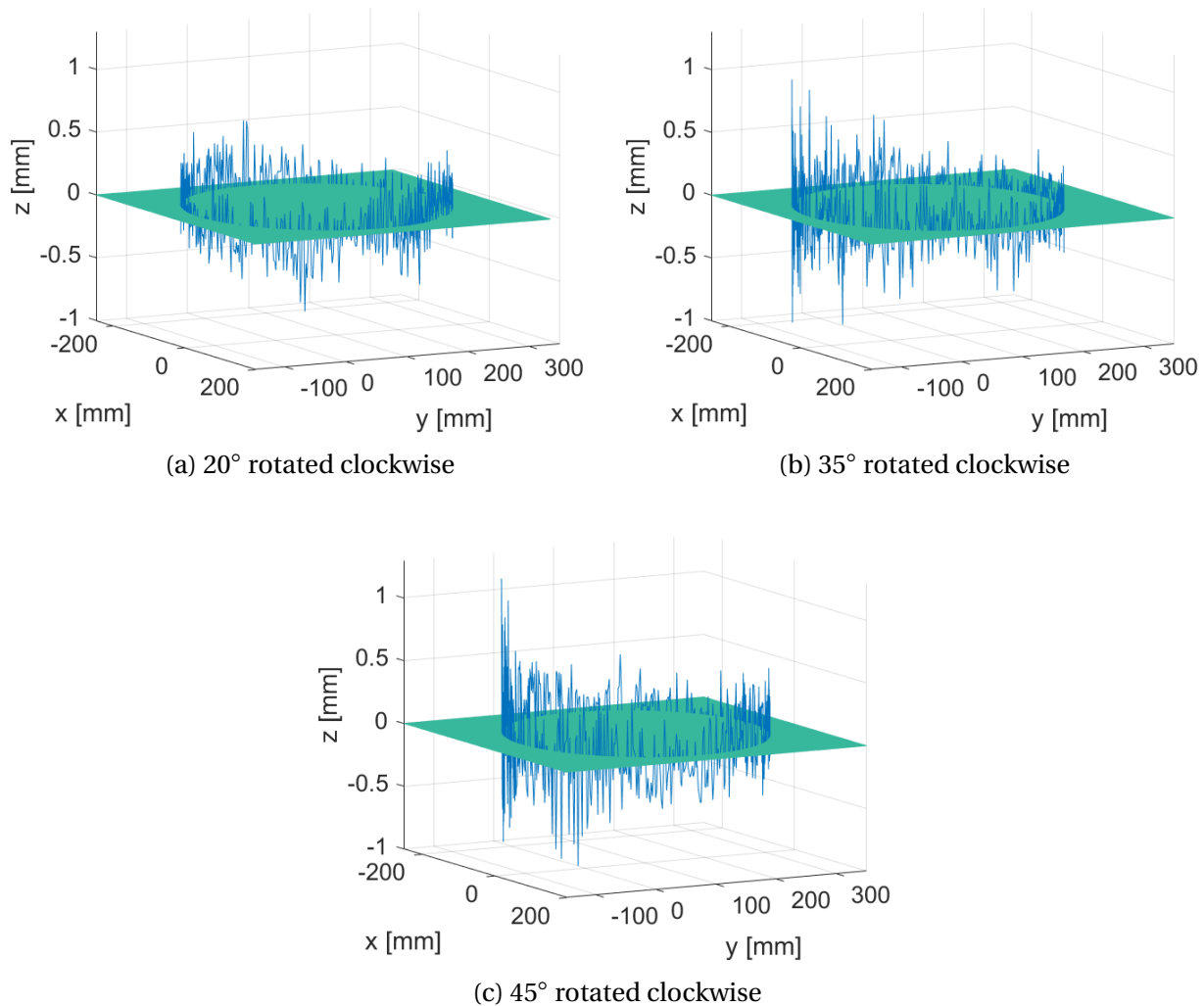


Figure 4.19: Fluctuating depth reconstruction of full rotation tracking

Fluctuations of depth reconstructions are plotted in figure 4.19. Visually, the errors of Center 1 are showing best results. The extremal values of figure 4.19a are -0.56 and 0.64 mm, being smaller than the spikes seen in figures 4.19b and 4.19c of magnitudes close to ± 1 mm. The mean and standard deviations provided in table 4.4 on the contrary, are very similar. Circle 1 and 2 have means separated by only 0.01 mm, and standard deviations separated by 0.03 mm. Circle 3 are estimating largest reconstruction errors in both radius and depth reconstructions.

4.3.4 Discussion of full rotation tracking

The results from the reconstruction of the rotational path of the plank shows good radius estimation in all sequences. The worst predicted mean radius, done by Circle 3, was only 98.92% wrongly estimated. Expectingly, this error occurs when the plane is rotated most, i.e. 45° . Reasons for this can be caused by the circle detection struggling to detect the exact circle when viewed from an angle. The tracking point viewed from the one camera, will be skewed and not completely circular as showed in figure 4.20. The circle detection code estimate the center point, hence, the possibility of miscalculation of the point may increase when the point is skewed. Similar phenomena can also be found in depth estimation in figures 4.19b and 4.19c. The fluctuations are more extreme at low values in y -direction. When y has smallest value, the plank is located as showed in figure 4.20, and the tracking point resembles an ellipse more than

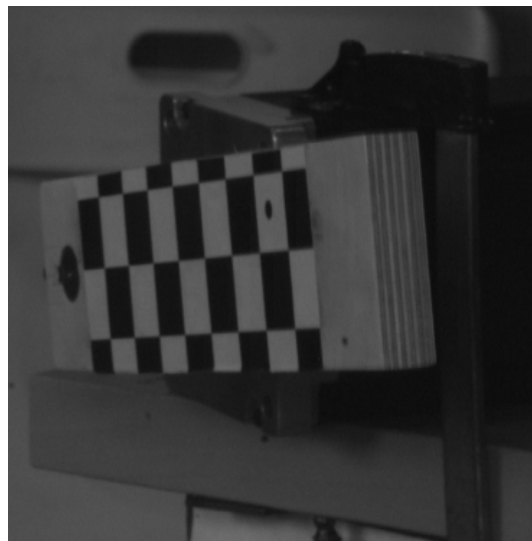


Figure 4.20: Illustration of skewed tracking point when viewed from angle

a circle.

Enhancement of this error can be done by using different recognition tools for detecting the tracking point. In PIV and PTV, the particles are found by scanning through each image pixel and detect intensities higher or lower than a certain threshold. Applying this method to this thesis' approach requires the surroundings of the object to be a solid color without objects of equal intensity as the point intensity.

Furthermore, similar discussion as for the arc tracking can be applied for the full rotation tracking. Applying expected calibration errors from section 4.1 to each directions enhances the results slightly. However, the extremal values are still quite high, and other causes may be of more significance, as for instance the circle detection discussed above. Additionally, the physical errors due to wrongly measured distances of real radius, can be questioned if are of more importance in the full rotation results than the arc results. All of the three reconstructions of radii spans over 0.4 mm from 192.9 to 193.3 mm. If 0.5 mm error can be expected from each end of the distance measured, a total of 1 mm can have been wrongly measured. Better tools than a measuring band could have been used in order to reduce this possible source of error.

Comparing the accuracy of the arc tracking and the full rotation tracking, the arc tracking are showing better results for the radii estimation. The mean depth errors does not differ significantly. However, extremal values illustrated in the plots of depth fluctuations, indicates a larger variations of depth in the full rotation reconstruction. To establish how an angled object may cause less accurate results, a full rotation experiment parallel to the camera traverse should have been performed.

4.4 Uncertainty analysis

This thesis, may in general, be interpreted as an uncertainty analysis of this particular tracking procedure with cameras. All the experiments conducted are to evaluate accuracy, and the numerical analysis is performed to establish an optimal camera setup. However, there will still be some errors related to the equipments used, and human errors related to for example measuring. Some sources of error are already mentioned in the previous discussion sections, but will be included here to get an overview of all uncertainties relevant for the results.

The sources related to reconstructing the path of an object with cameras can be

- Wrongly measured distances of radius. This will cause the error of the reconstructed radii to be higher or lower than the actual error.
- Badly recognised tracking point in image frames. The center point can be slightly wrong, causing an error of $n \times \text{wrong pixels} \times \frac{[\text{mm}]}{\text{pixel}}$.
- Movements of camera traverse during imaging. The traverse was located at a table next to the computer, which was vibration-sensitive to movements close to the table.
- Reconstruction errors related to individual calibrations. Expected errors from calibration are often individual. The results from the first calibration experiment in section 4.1 are therefore guiding, but not completely representative for each tracking sequence.

Chapter 5

Conclusion

The work of the thesis are presented in the earlier chapters. In this last chapter, the most important findings are repeated, and a conclusion is drawn from the objectives stated in the introduction. Additionally, suggestions for further work are thereafter given, meant as a possible continuation of the work conducted in this thesis.

5.1 Conclusions

The path of a rotating object has been tracked in the 3D world space. Its circular trajectories have been reconstructed and analysed to estimate the accuracy of the tracking. A camera calibration experiment was performed to estimate expected reconstruction errors that may be caused due to inaccuracies of the calibration. To determine the sensitivity of positioning the cameras used in the tracking procedure, a numerical investigation was performed. Both number of cameras to use and the angle between them, were numerically studied. Last, tracking experiments of different setups were conducted to establish the accuracy of reconstructing the trajectory and the depth variation of the path of a rotating plank.

Both image capturing methods of the camera calibration experiment predicted approximately equal reconstruction errors of mean and standard deviations. The errors, resulted in 0.05 mm in x -direction, 0.04 mm in y -direction, and 0.1 mm in z -direction. The error in z -direction was obviously the largest error, coinciding with literature study of calibration accuracy [22]. Small errors are therefore to expect in all directions when reconstructing world points from

image coordinates. To decrease the sensitivity of the depth estimation, using 3D calibration object is suggested.

The numerical study of finding the optimal camera positioning in space, by varying the number of cameras and the angle between them, resulted in smallest errors of an asymmetric four camera setup as depicted in figure 3.7. However, insignificant differences in error values was seen for the symmetric four camera and symmetric three camera setups, coinciding with results from Maas [12]. Expectingly, using two cameras gave highest errors. However, the asymmetric three camera setup resulted in almost as bad errors, concluding with applying a symmetric setups when only three cameras are available. Furthermore, all of the camera setups resulted in an optimal angle $\alpha = 45^\circ$. The majority of the results did not depend on θ , implying the positioning in xz -plane, or vertical direction, being of most importance.

Results from the tracking experiment obtained good estimations of the shape of the rotating trajectory. The radius of the arc tracking path being parallel to the camera traverse estimated the radius within 99.4% of the actual path radius. Turning the rotational plane away from the parallel plane 20° , 35° and 45° resulted in radius estimation accuracy of 99.1%, 99.1% and 98.9%, respectively. The depth fluctuations of the two experiments did not differ significantly in mean values. However, from the plots, the scale in z -direction implied higher extremal values when reconstructing the full rotation. Best results are therefore to be expected in plane rotation parallel to the camera traverse. Reasons for the fluctuations of depth caused by vibrations, can not be concluded with from these experiments. Arc 2, sequence (a) which was the non-vibrating series, did not result in significantly different mean errors or standard deviations in z -direction compared with Arc 2 (b). Nevertheless, closer investigation of the vibrations of the plank caused by the stepper motor should be conducted as further work.

Applying the knowledge to the water tank facility experiment, one can conclude with the use of four cameras being optimal and resulting in best accuracy. Locating the cameras in a symmetric or an asymmetric setup, will not change the accuracy significantly. The angle in vertical direction, however, should be 45° , and the calibration of the cameras should include images of the checkerboard at different orientations covering also the corners of camera frames.

5.2 Recommendations for Further Work

In this section, suggestions for further work are presented. The suggestions are related to the work that has been conducted in this thesis, and are either enhancement of methods or continued study with the applied methods.

Recalling the aim of the thesis being to reconstruct the trajectories of a rotating object using cameras. This objective had the purpose of being applied to a future water tank experiment in the Fluids Department at NTNU. How the work from this thesis can be applied to this experiment are elaborated in upcoming section [5.2.1](#).

Furthermore, suggestions for other experimental approaches of the tracking procedure are proposed to obtain better accuracy and minimize the uncertainties related to the analysis. These recommendations are provided in section [5.2.2](#).

5.2.1 Water tank experiment

The natural extension of the thesis is to perform the water tank experiment planned in the Fluids Department at NTNU. Unfortunately, time for construction of the tank exceeded the time of this master thesis, and the planned experiments in the tank was not performed. However, objectives for this experiment are suggested to be

- Track the helical path of the rotating turbine blades
- Identify vibrations or uneven trajectories of the turbine
- Visualize the wing tip vortices and the evolution of the wake

The experimental facility will become as depicted in figure [5.1](#). It consists of two main parts: the water tank; and the traverse with two stepper motors to rotate and translate the turbine blades. The tank has a cylindrical shape, but consists of 12 vertical Plexiglas plates glued properly together around the baseplate. This implies an angle of 30° between the views orthogonal to the 12 wall plates, and towards the center of the tank. This information is necessary in order to locate the cameras around the tank without compensating for radial shift in multimedia (air, Plexiglas and water) geometry [[12](#)]. Since camera angles solely was dependent on the angle in

the vertical direction, positioning the cameras orthogonal to the glass plates should not affect the accuracy of the tracking.

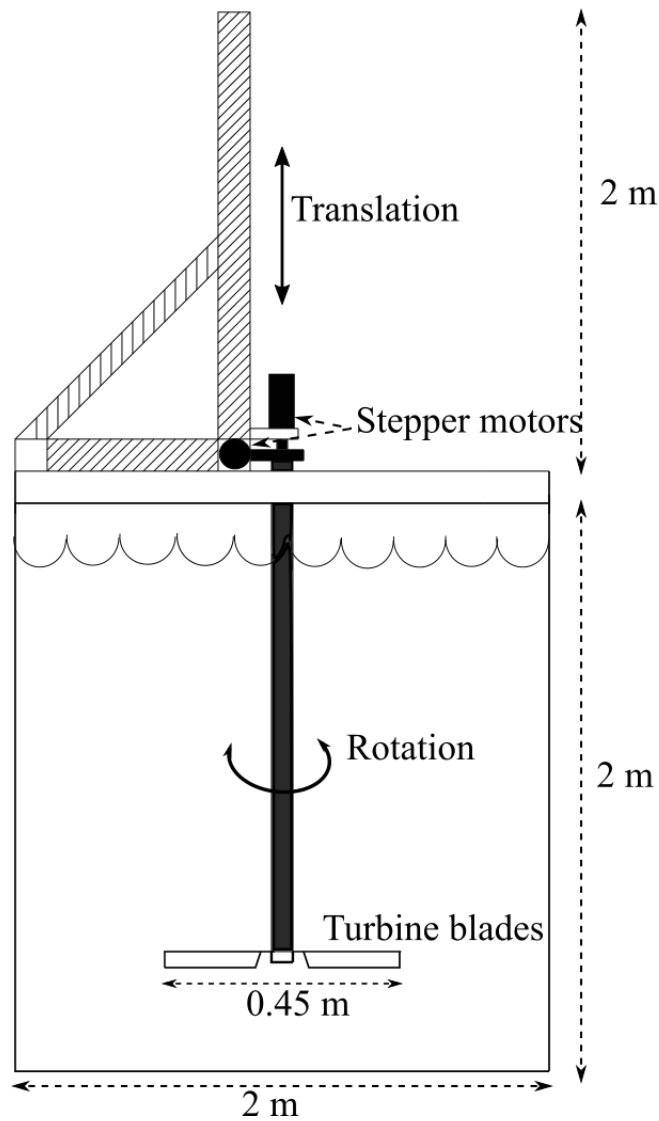


Figure 5.1: Experimental Towing tank rig

Primarily, preparations already conducted to run the water tank experiments, but not used in the thesis, are

- Writing a risk assessment report
- Writing a Labview script for controlling the stepper motors

Moreover, codes for camera calibration and tracking of a point in an image sequence has been

written. The codes are used in the experimental tracking of this thesis, and are provided in Appendix section C.1 and C.3.

The risk assessment report can be used further as a guideline for creating the new risk report by the next user. Relevant parts of the report are provided in Appendix section D, and contains evaluation of the risks assessed as important for the water tank facility in figure 5.1. Controlling the stepper motors can be done with Labview through a DAQ-device with both digital input-outputs and counters. The Labview script providing signals for the stepper motor controller are provided in Appendix section C.4.

These preparations are only guidelines of how this experiment can be done, and no solution manual. For instance, a new risk assessment must be conducted by the next scientist, and the codes written may not be suitable for all applications.

Track helical path of turbine blades

The first objective suggested is reconstructing the helical path of rotating and translated turbine blades. This implies applying the same method as for tracking a point of a rotating object described in section 3.3, in terms of the calibration and tracking code. The number of cameras to be used and how they should be located, can be interpreted from the results in chapter 4. Having four cameras available is therefore optimal in terms of accuracy, and locating them with a vertical angle $2 \times \alpha$ of 90° between them will result in the most accurate tracking results. Connecting the stepper motors to a computer and controlling them with the Labview script would be the next step.

Regarding the turbine blades, it would be interesting to test the procedure for actual wind turbine blades. However, starting by assembling simplified flat blades are a good way to check the tracking method and setup of the rig. Real turbine blades can be tested when the simplified ones shows good results of reconstructing their trajectories.

Identify vibrations of turbine blades

The second objective is to identify possible vibrations of the turbine blades in vertical direction. This phenomena will occur if the turbine blades are of a flexible material, or if the rod connecting the blade and the traverse moves in horizontal direction. For this to be measured, cameras

can be placed to capture one image frame of a point on the blades at every rotation. Being able to capture the frame at the same blade position, the rotational speed of the turbine is necessary.

Visualize wing tip vortices and wake evolution

The previous objective can be connected to why visualization of the wing tip vortices are of interest. If vibrations of the blades are found, investigation of how these vibrations affects the wake evolution is an interesting topic. A method for how the visualization easily can be performed is by having dye exiting the tips of the blades. The dye must be neutrally buoyant, and will then follow the flow downstream of the blades and show the flow of the wake. Limitations by using this visualization method is that the tank holds a volume of about 6000 litres of water, that must be drained between each experimental sequence.

5.2.2 Tracking enhancement

Enhancement of point tracking will mostly be dependent on the camera calibration and their setups. Therefore, objectives to be studied further in terms of accuracy are

- verification of the numerical accuracy analysis of camera positioning
- perform camera calibration with a 3D calibration object
- estimate vibrations of tracking object

Due to limitations of cameras available in the experiments, the different camera setups numerically investigated were not verified experimentally. Conducting an experiment with the camera orientations of three and four - symmetric and asymmetric - camera setups can be done to verify the numerical analysis. These experiments depends on camera availability and variable scaffolds to mount the cameras onto.

Performing the camera calibration with 3D calibration objects may be performed to enhance the accuracy of the calibration slightly, see [22]. An object with three dimensional geometry are therefore necessary. These objects are expensive to make, and not as easily applicable as the checkerboard used in this thesis.

To neglect errors due to instrumental causes, a procedure of evaluating the vibrations of the apparatus can be conducted. This can be done, as presented in section 4.3.2, by imaging the tracking object from the side. If a sideways tracking is performed, it will change the coordinate system primarily being the xz -plane, to be the xy -plane. Earlier mentioned literature of Samper et al. [22] states smaller reconstruction errors in x and y -directions, making the predictions of the vibrations more accurate from the side. In this experiment, capturing more than 50 fps could be necessary to investigate the movements thoroughly.

Bibliography

- [1] Alfredson, P. and Dalhberg, J.-A. (1979). A preliminary wind tunnel study of windmill wake dispersion in various flow conditions. *Technical Note AI-1499*.
- [2] Bucher, I. (2004). Circle fit. <http://www.mathworks.com/matlabcentral/fileexchange/5557-circle-fit>. [Online; accessed 20-May-2016].
- [3] Faugeras, O. (1993). *Three-Dimensional Computer Vision*. MIT Press.
- [4] Fillmore, J. P. (1984). A note on rotation matrices. *IEEE Computer Graphics and Applications*, 4(2):30–33.
- [5] Guezennec, Y. G., Brodkey, R. S., Trigui, N., and Kent, J. C. Algorithms for fully automated three-dimensional particle tracking velocimetry. *Experimentals in Fluids* 17, 209-219.
- [6] Homeniuk, D. L., Nobes, D. S., and Wilson, D. J. (2010). Three-dimensional particle tracking velocimetry. *The Canadian Society for mechanical Engineering. University of Alberta. T6G 2G8*.
- [7] Illingworth, J. and Kittler, J. (1988). A survey of the hough transform. *Computer vision, graphics, and image processing*, 44(1):87–116.
- [8] Krause, E. (2007). Barrel distortion. http://wiki.panotools.org/index.php?title=Barrel_distortion&oldid=9378. Online; accessed 20-may-2016.
- [9] Langer, M. (2010). Lecture notes: Camera rotation, homogeneous coordinates. <http://www.cim.mcgill.ca/~langer/558/3-rotation.pdf>.

- [10] Lawson, N. J. and Wu, J. Three-dimensional particle image velocimetry: experimental error analysis of a digital angular stereoscopic system. *Meas. Sci. Technol.* 8, 1455-1464.
- [11] Lawson, N. J. and Wu, J. (1997). Three-dimensional particle image velocimetry: error analysis of stereoscopic techniques. *Meas. Sci. Technol.* 8, 894-900.
- [12] Maas, H. G., Gruen, A., and Papantoniou, D. (1993). Particle tracking velocimetry in three-dimensional flows - part 1. photogrammetric determination of particle coordinates. *Experimental in Fluids* 15, 133-146.
- [13] Mann, J., Ott, S., and Andersen, J. S. (1999). Experimental study of relative, turbulent diffusion. *Risø-R-1036(EN)*.
- [14] Manwell, J., McGowan, J., and Rogers, A. (2009). *Wind Energy Explained: Theory, Design and Application*. John Wiley and Sons, Ltd.
- [15] Marcato Junior, J., Moraes, M. V. A. d., and Tommaselli, A. M. G. (2015). Experimental assessment of techniques for fisheye camera calibration. *Boletim de Ciências Geodésicas*, 21(3):637–651.
- [16] MathWorks. Single camera calibration app. <http://se.mathworks.com/help/vision/ug/single-camera-calibrator-app.html>. [Updated for R2016a].
- [17] MathWorks (2012). Image processing toolbox, `imfindcircles`. http://se.mathworks.com/help/images/ref/imfindcircles.html#outputarg_metric.
- [18] Medioni, G. and Kang, S. B. (2004). *Emerging Topics in Computer Vision, Ch 2, pages 4-43*. Prentice Hall Professional Technical Reference.
- [19] Morvan, Y. (2009). Aquisition, compression and rendering of depth and texture for multi-view video. <http://www.epixea.com/research/multi-view-coding-thesis.html>. [Chapter 2, Projective Geometry].
- [20] Nemes, A., Sherry, M., Jacono, D. L., Blackburn, H., and Sheridan, J. (2014). Evolution and breakdown of helical vortex wakes behind a wind turbine. In *Journal of Physics: Conference Series*, volume 555, page 012077. IOP Publishing.

- [21] Ryberg, A. (2010). *Camera Modelling and Calibration with Machine Vision Applications*. PhD thesis, Chalmers University of Technology.
- [22] Samper, D., Santolaria, J., Brosted, F. J., Majarena, A. C., and Aguilar, J. J. (2013). Analysis of tsai calibration method using two-and three-dimensional calibration objects. *Machine vision and applications*, 24(1):117–131.
- [23] Sanderse, B. (2009). Aerodynamics of wind turbine wakes. *Energy Research Center of the Netherlands (ECN), ECN-E-09-016, Petten, The Netherlands, Tech. Rep*, 5(15):153.
- [24] Sawyer, S. and Rave, K. (2014). Global wind report 2014. Technical report, Global Wind Energy Council (GWEC).
- [25] SmartMotorDevices (2014). Programmable step motor controller smsd-822. <http://wpstatic.idium.no/aratron.no/2015/01/Smartmotor-SMSD-8-22-steppmotorstyring.pdf>.
- [26] Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4).
- [27] Vermeer, L., Sørensen, J., and Crespo, A. (2003). Wind turbine wake aerodynamics. *Progress in Aerospace Science* 39, (39):467–510.
- [28] Wolfram, M. Spherical coordinates. <http://mathworld.wolfram.com/SphericalCoordinates.html>. [Online; accessed 8-February-2016].
- [29] Yun, X., Bachmann, E. R., Kavousanos-Kavousanakis, A., Yildiz, F., and Mcghee, R. B. (2004). Design and implementation of the marg human body motion tracking system. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 625–630. IEEE.
- [30] Zhang, Z. (2000). A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(11):1330–1334.

Appendix A

Additional Information

Additional information of theory is provided in this chapter.

A.1 Derivation of equation 2.15 in section 2.3

Starting with equation 2.14 and applying the relationship between the camera center and translation vector and rotation matrix given in equation 2.11, we can write

$$\lambda \tilde{\mathbf{m}} = \mathbf{P} \tilde{\mathbf{M}} = \mathbf{A}[\mathbf{R} \ \mathbf{t}] \tilde{\mathbf{M}} = \mathbf{A}[\mathbf{R} \ -\mathbf{RC}] \tilde{\mathbf{M}}$$

writing $\tilde{\mathbf{M}} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$, we get

$$\lambda \tilde{\mathbf{m}} = \mathbf{A}[\mathbf{R} \ -\mathbf{RC}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Since \mathbf{R} is a 3 by 3 matrix and $-\mathbf{RC}$ is a 3 by 1 matrix, it follows

$$\lambda \tilde{\mathbf{m}} = \mathbf{AR} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{ARC}$$

Multiplying with the $\mathbf{R}^{-1}\mathbf{A}^{-1}$ on both sides of the equation

$$\mathbf{R}^{-1}\mathbf{A}^{-1}\lambda \tilde{\mathbf{m}} = \mathbf{R}^{-1}\mathbf{A}^{-1}\mathbf{AR} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{R}^{-1}\mathbf{A}^{-1}\mathbf{ARC} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - \mathbf{C}$$

which leads to the wanted equation 2.15 for the world point

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \mathbf{C} + \mathbf{R}^{-1} \mathbf{A}^{-1} \lambda \mathbf{\bar{m}}$$

A.2 Setting parameters of `imfindcircles`

- circle radius span [r_{min} r_{max}]. This makes the function only search for circles with radius within the minimum and maximum value.
- `ObjectPolarity` must be set to determine whether the circle is bright or dark in comparison with the image background.
- `Sensitivity`. If the circle can be difficult to detect, the `Sensitivity` of detection can be increased. This is a value between 0 and 1, where a high value may detect more circular objects. Setting a high `Sensitivity` may also result in false circle detections.
- `computationalMethod` can be chosen between two different types; Atheron and Kerbyson's `PhaseCode` method, or the method used in `TwoStage` circular Hough transform.
- `EdgeThreshold`, which sets the gradient threshold for determining the edge pixels in the image. The value can be set between 0 and 1, where a lower threshold value detects more circles, and vice versa.

Appendix B

Numerical parameters

In the following tables, all the rotation matrices used in the numerical angle analysis are listed. Only the roll, pitch and yaw matrices are given, while the total rotation matrix for the camera is the product of all three rotational directions in following order:

$$R_{tot} = R_{yaw} R_{pitch} R_{roll}$$

The camera positions are also presented as x , y and z -coordinates.

Table B.1: Rotation matrices and camera coordinates for numerical angle study of two cameras

CAMERA NO	CAMERA COORDINATES	ROTATION MATRIX
<i>Camera 1</i>	$x = r \sin(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$
	$y = r \sin(\theta) \cos(-\alpha)$	$R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$
	$z = r \cos(\theta) \cos(-\alpha)$	$R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 2</i>	$x = r \sin(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{bmatrix}$
	$y = r \sin(-\theta) \cos(-\alpha)$	$R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$
	$z = r \cos(-\theta) \cos(-\alpha)$	$R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table B.2: Rotation matrices and camera coordinates for numerical angle study of three cameras located in a symmetric triangle

CAMERA NO	CAMERA COORDINATES	ROTATION MATRIX
<i>Camera 1</i>	$x = r \sin(\alpha)$ $y = 0$ $z = r \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(0) & -\sin(0) \\ 0 & \sin(0) & \cos(0) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 2</i>	$x = r \sin(-\alpha)$ $y = r \sin(\theta) \cos(-\alpha)$ $z = r \cos(\theta) \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 3</i>	$x = r \sin(-\alpha)$ $y = r \sin(-\theta) \cos(-\alpha)$ $z = r \cos(-\theta) \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table B.3: Rotation matrices and camera coordinates for numerical angle study of three cameras located in an asymmetric triangle

CAMERA NO	CAMERA COORDINATES	ROTATION MATRIX
<i>Camera 1</i>	$x = r \sin(\alpha)$ $y = 0$ $z = r \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(0) & -\sin(0) \\ 0 & \sin(0) & \cos(0) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 2</i>	$x = r \sin(-\alpha)$ $y = 0$ $z = r \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(0) & -\sin(0) \\ 0 & \sin(0) & \cos(0) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 3</i>	$x = 0$ $y = r \sin(\theta)$ $z = r \cos(\theta)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(0) & 0 & \sin(0) \\ 0 & 1 & 0 \\ -\sin(0) & 0 & \cos(0) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table B.4: Rotation matrices and camera coordinates for numerical angle study of four cameras located symmetrically

CAMERA NO	CAMERA COORDINATES	ROTATION MATRIX
<i>Camera 1</i>	$x = r \sin(\alpha)$ $y = r \sin(\theta) \cos(\alpha)$ $z = r \cos(\theta) \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 2</i>	$x = r \sin(-\alpha)$ $y = r \sin(\theta) \cos(-\alpha)$ $z = r \cos(\theta) \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 3</i>	$x = r \sin(\alpha)$ $y = r \sin(-\theta) \cos(\alpha)$ $z = r \cos(-\theta) \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 4</i>	$x = r \sin(-\alpha)$ $y = r \sin(\theta) \cos(-\alpha)$ $z = r \cos(\theta) \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(-\theta) & -\sin(-\theta) \\ 0 & \sin(-\theta) & \cos(-\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table B.5: Rotation matrices and camera coordinates for numerical angle study of four cameras located asymmetrically

CAMERA NO	CAMERA COORDINATES	ROTATION MATRIX
<i>Camera 1</i>	$x = r \sin(\alpha)$ $y = 0$ $z = r \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(0) & -\sin(0) \\ 0 & \sin(0) & \cos(0) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 2</i>	$x = r \sin(-\alpha)$ $y = 0$ $z = r \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(0) & -\sin(0) \\ 0 & \sin(0) & \cos(0) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 3</i>	$x = r \sin(\alpha)$ $y = r \sin(\theta) \cos(\alpha)$ $z = r \cos(\theta) \cos(\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(\alpha) & 0 & \sin(\alpha) \\ 0 & 1 & 0 \\ -\sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$
<i>Camera 4</i>	$x = r \sin(-\alpha)$ $y = r \sin(\theta) \cos(-\alpha)$ $z = r \cos(\theta) \cos(-\alpha)$	$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}$ $R_{pitch} = \begin{bmatrix} \cos(-\alpha) & 0 & \sin(-\alpha) \\ 0 & 1 & 0 \\ -\sin(-\alpha) & 0 & \cos(-\alpha) \end{bmatrix}$ $R_{yaw} = \begin{bmatrix} \cos(0) & -\sin(0) & 0 \\ \sin(0) & \cos(0) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Appendix C

MATLAB and Labview scripts

MATLAB and Labview scripts used in the thesis are here provided.

C.1 MATLAB camera calibration code

Following matlab code used for camera calibration:

```
1 %Camera calibration with a 2D object.
2 clear all; close all;
3 for i = numImages
4     imageName = sprintf('left%d000001.tif',i);
5     imageNames{i} = fullfile('\imageFolder', imageName);
6 end
7 %General parameters:
8 squareSize = 25;%size of the checkerboard squares
9 %Extract calibration points and board size from the different images.
10 [iP, boardSize, imagesUsed] = detectCheckerboardPoints(imageNames); ...
    %detect pattern and estimate u and v from image
11 imageNames = imageNames(imagesUsed);
12 %Generate known points corresponding to the calibration points in the world
13 %coordinate system.
14 wP = generateCheckerboardPoints(boardSize, squareSize);
15 calibrationPoints = size(iP,1); %how many points used for the calibration
```

```

16 [cameraParams, imagesUsed, estimationErrors] = ...
    estimateCameraParameters(iP, wP, ...
17     'EstimateSkew', false, 'EstimateTangentialDistortion', false, ...
18     'NumRadialDistortionCoefficients', 2, 'WorldUnits', 'mm');
19 R_transposed = cameraParams.RotationMatrices; %extract rotation ...
    matrices from camera parameters
20 t = cameraParams.TranslationVectors; %extract translation vectors
21 A = cameraParams.IntrinsicMatrix; %extract instrinsic parameters
22 numImages = size(imagesUsed,1);
23 [imagePoints, worldPoints, A, t, R, P] = fixParameters(iP, wP, ...
    A,t,R_transposed, numImages, calibrationPoints);

```

Find parameters for all camera locations. Determine the lines of sight and calculate intersection point:

```

1 %% Determine the deviation from original point.
2
3 Δ = zeros(calibrationPoints, numImages);
4 pointCoordinates2 = zeros(3,calibrationPoints, numImages);
5 Δx = zeros(calibrationPoints, numImages);
6 Δy = zeros(calibrationPoints, numImages);
7 Δz = zeros(calibrationPoints, numImages);
8 for i = 1:numImages
9     for j = 1:calibrationPoints
10        imagePoint_left2 = imagePoints_left(:,j,i); %pixel value of ...
            point from left camera
11        imagePoint_right2 = imagePoints_right(:,j,i); %pixel value of ...
            point from right camera
12        imagePoint_middle2 = imagePoints_middle(:,j,i); %pixel value of ...
            point from middle camera
13        [worldPosition_left2, C_left2] = calculateLinesOfsight(A_left, ...
            R_left(:, :, i), t_left(:, i), imagePoint_left2);
14        [worldPosition_right2, C_right2] = ...
            calculateLinesOfsight(A_right, R_right(:, :, i), t_right(:, i), ...
            imagePoint_right2);

```

```

15     [worldPosition_middle2, C_middle2] = ...
        calculateLinesOfsight (A_middle, R_middle(:, :, i), ...
            t_middle(:, i), imagePoint_middle2);
16     pointCoordinates2(:, j, i) = ...
        detectIntersectionPoint (worldPosition_left2, ...
            worldPosition_right2, worldPosition_middle2);
17     Δx(j, i) = (wP(j, 1) - pointCoordinates2(1, j, i));
18     Δy(j, i) = (wP(j, 2) - pointCoordinates2(2, j, i));
19     Δz(j, i) = (0 - pointCoordinates2(3, j, i));
20     Δ(j, i) = norm([wP(j, :)'; 0] - pointCoordinates2(:, j, i));
21     end
22 end

```

C.2 MATLAB numerical analysis code

Code used in numerical angle analysis

```

1 close all; clear all;
2 %Monte carlo of angle study
3 t2 = [15 30 45 60];
4 a = [10 15 20 25 30 35 40 45 50 55 60];
5
6 for l = 1:length(t2)
7     for h = 1:length(a)
8         alpha1 = a(h);
9         theta1 = t2(l);
10
11         for k = 1:1000
12             Numerical_angle3cam
13         end
14     end
15 end
16
17 %This function/script conducts a numerical study of diffetent angles

```

```

18 %between cameras on a traverse. It needs a camera matrix, as the intrinsic
19 %parameters of a camera, and then the extrinsic parameters, as the rotation
20 %and camera position, are varied to simulate cameras at different angles.
21
22 A = [5386, 0, 494.6; 0, 5381, 444.5; 0, 0, 1];
23
24 %Generate checkerboard world points as reference points
25 row = [0 25 50 75 100 125];
26 row2 = [0 25 50 75 100 125 150];
27 for i = 1:7
28     worldPoints(((i-1)*6+1):((i-1)*6+6),1) = row2(i);
29     worldPoints(((i-1)*6+1):((i-1)*6+6),2) = row';
30 end
31
32 numberOfCameras = 4;
33
34 %Need to define radius and angles to get camera position:
35 r = -1000; %1000mm from object, all cameras
36 alpha = alphas;%45;
37 theta = thetas;
38
39 %convert to radians
40 alpha_rad = alpha*(pi/180);
41 theta = theta*(pi/180);
42
43 %Define rotation matrices for cameras, must be changed for each camera ...
44     setup:
45 R = zeros(3,3,numberOfCameras);
46 C = zeros(3,numberOfCameras);
47
48 %pitch = y-axis, roll = x-axis, yaw = z-axis
49 %cam 1:
50 R_roll1 = [1, 0, 0; 0, cos(theta), -sin(theta); 0, sin(theta), cos(theta)];
51 R_pitch1 = [cos(alpha_rad), 0, sin(alpha_rad); 0, 1, 0; ...
52     -sin(alpha_rad), 0, cos(alpha_rad)];
53 R_yaw1 = [cos(0), -sin(0), 0; sin(0), cos(0), 0; 0, 0, 1];

```



```
52 R(:,:,1) = R_yaw1*R_pitch1*R_roll1;
53
54 %cam 2:
55 R_roll2 = [1, 0, 0; 0, cos(theta), -sin(theta); 0, sin(theta), cos(theta)];
56 R_pitch2 = [cos(-alpha_rad), 0, sin(-alpha_rad); 0, 1, 0; ...
    -sin(-alpha_rad), 0, cos(-alpha_rad)];
57 R_yaw2 = [cos(0), -sin(0), 0; sin(0), cos(0), 0; 0, 0, 1];
58 R(:,:,2) = R_yaw2*R_pitch2*R_roll2;
59
60 %cam 3:
61 R_roll3 = [1, 0, 0; 0, cos(-theta), -sin(-theta); 0, sin(-theta), ...
    cos(-theta)];
62 R_pitch3 = [cos(alpha_rad), 0, sin(alpha_rad); 0, 1, 0; ...
    -sin(alpha_rad), 0, cos(alpha_rad)];
63 R_yaw3 = [cos(0), -sin(0), 0; sin(0), cos(0), 0; 0, 0, 1];
64 R(:,:,3) = R_yaw3*R_pitch3*R_roll3;
65
66 %cam 4:
67 R_roll4 = [1, 0, 0; 0, cos(-theta), -sin(-theta); 0, sin(-theta), ...
    cos(-theta)];
68 R_pitch4 = [cos(-alpha_rad), 0, sin(-alpha_rad); 0, 1, 0; ...
    -sin(-alpha_rad), 0, cos(-alpha_rad)];
69 R_yaw4 = [cos(0), -sin(0), 0; sin(0), cos(0), 0; 0, 0, 1];
70 R(:,:,4) = R_yaw4*R_pitch4*R_roll4;
71
72 %camera position must be changed for each setup
73 %camera 1:
74 x(1) = r*sin(alpha_rad);
75 y(1) = r*sin(theta)*cos(alpha_rad);
76 z(1) = r*cos(theta)*cos(alpha_rad);
77
78 %camera 2:
79 x(2) = r*sin(-alpha_rad);
80 y(2) = r*sin(theta)*cos(-alpha_rad);
81 z(2) = r*cos(theta)*cos(-alpha_rad);
82
```

```
83 %camera 3:
84 x(3) = r*sin(alpha_rad);
85 y(3) = r*sin(-theta)*cos(alpha_rad);
86 z(3) = r*cos(-theta)*cos(alpha_rad);
87
88 %camera 3 for 4 cameras:
89 x(4) = r*sin(-alpha_rad);
90 y(4) = r*sin(-theta)*cos(-alpha_rad);
91 z(4) = r*cos(-theta)*cos(-alpha_rad);
92
93 C = [x(1,:); y(1,:); z(1,:)];
94
95 for i = 1:numberOfCameras
96     %Translation vector
97     t(:,i) = -R(:, :, i)*C(:, i);
98     %Projection matrix
99     P(:, :, i) = A*[R(:, :, i) t(:, i)];
100 end
```

C.3 MATLAB point tracking code

Code for tracking after loading images and calibrating cameras:

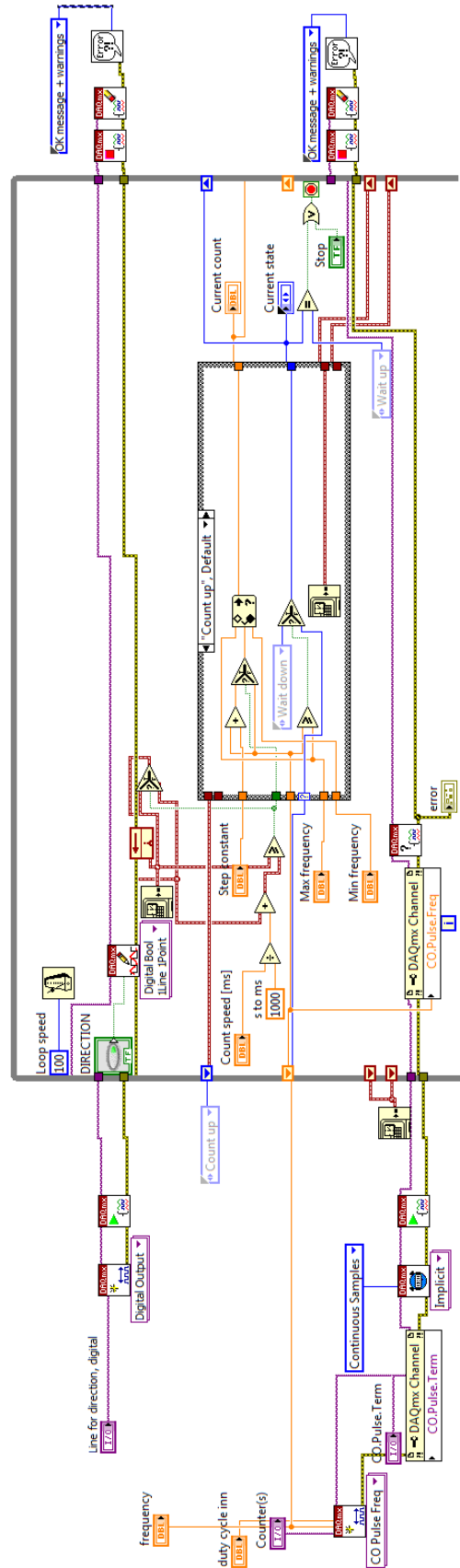
```
1 %Detect circle in image
2 %Need to find approximate circle radius to make the detection faster
3 figure(1)
4 imshow(images1{1})
5 d = imdistline;
6 %Measure the radius of the point in image (in pixels)
7 %delete(d);
8 min_radius = 'What is the minimum radius?';
9 max_radius = 'What is the maximum radius?';
10 start1 = 'Give coordinates of startpoint 1: ';
11 r1 = input(min_radius); %set minimum radii
```

```
12 r2 = input(max_radius); %set maximum radii
13 startPoint1 = input(start1);
14 figure(2)
15 imshow(images2{1})
16 start2 = 'Give coordinates of startpoin 2: ';
17 startPoint2 = input(start2);
18 %Detect circles:
19 Centers1 = zeros(nf,2);
20 Centers2 = zeros(nf,2);
21 Radius1 = zeros(nf,1);
22 Radius2 = zeros(nf,1);
23 for i = 1:nf
24     [center1, radii1] = imfindcircles(images1{i}, [r1 ...
25         r2], 'ObjectPolarity', 'dark', ...
26         'Sensitivity', 0.86, 'Method', 'tstage');
27     [center2, radii2] = imfindcircles(images2{i}, [r1 ...
28         r2], 'ObjectPolarity', 'dark', ...
29         'Sensitivity', 0.87, 'Method', 'tstage');
30     for j = 1:size(center1,1)
31         if (norm(center1(j,:) - startPoint1) < 8 && center1(j,1) ≠ 0)
32             Centers1(i,:) = center1(j,:);
33             Radius1(i) = radii1(j);
34             startPoint1 = Centers1(i,:);
35         end
36     end
37     for k = 1:size(center2,1)
38         if (norm(center2(k,:) - startPoint2) < 8 && center2(k,1) ≠ 0)
39             Centers2(i,:) = center2(k,:);
40             Radius2(i) = radii2(k);
41             startPoint2 = Centers2(i,:);
42         end
43     end
44 end
```

C.4 Labview script step motor

Controlling the stepper motors to be used for rotating the shafts in the water tank, is done through Labview. A PCI-6602 device from National Instruments is used to send square signals to the stepper motor controller. The PCI-6602 device has 8 up/down, 32-bit counters and 32 digital I/O lines.

The labview code starts and accelerate the motor, make it run on constant speed for a given time period, and then decelerate it before it stops. These different sequences are divided into different case structures. The block diagram of the code is provided below.



Appendix D

Risk assessment

Most relevant information from the conducted risk assessment are here provided.

ATTACHMENT B: HAZOP TEMPLATE

Project: Water tank with rotating turbine							Page
Node: 1							
Ref	Guideword	Causes	Consequences	Safeguards	Recommendations	Action	Date/Sign
	No flow	NA					
	Reverse flow	NA					
	More flow	NA					
	Less flow	NA					
	More level	<ul style="list-style-type: none"> Forget to turn off hose 	<ul style="list-style-type: none"> Water overflow Destroy electrical equipment 	<ul style="list-style-type: none"> Turn off water when leaving the room 	<ul style="list-style-type: none"> Stay with tank when filled Raise equipment off floor 		
	Less level	NA					
	More pressure	NA					
	Less pressure	NA					
	More temperature	NA					
	Less temperature	NA					
	More viscosity	NA					
	Less viscosity	NA					
	Composition Change	NA					
	Contamination	NA					
	Relief	NA					
	Instrumentation	NA					

Project: Water tank with rotating turbine						Page	
Node: 1							
Ref	Guideword	Causes	Consequences	Safeguards	Recommendations	Action	Date/Sign
	Sampling	NA					
	Corrosion/erosion	NA					
	Service failure						
	Abnormal Operation	<ul style="list-style-type: none"> The electric equipment does not work properly User error Stepping settings 	<ul style="list-style-type: none"> The shaft can be rotated outside maximum 	<ul style="list-style-type: none"> Pulse counter to stop when exceeds maximum 	<ul style="list-style-type: none"> Stay with the equipment when running the rotation and translation motors Perform calibration 		
	Maintenance	NA					
	Ignition	NA					
	Spare equipment	NA					
	Safety	<ul style="list-style-type: none"> Elevated equipment falling 	<ul style="list-style-type: none"> Equipment can damage persons Equipment can damage other objects 	<ul style="list-style-type: none"> Do not stand in falling zone 	<ul style="list-style-type: none"> Fasten equipment securely 		