

# A Model for an Augmented Reality Tool in Tumour Removal Laparoscopic Surgery

Master's thesis in Applied Physics

LISA MÅNSSON



MASTER'S THESIS 2020

# A Model for an Augmented Reality Tool in Tumour Removal Laparoscopic Surgery

LISA MÅNSSON



Department of Mathematical Sciences  
*Division of Applied Mathematics and Mathematical Statistics*  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2020

A Model for an Augmented Reality Tool in Tumour Removal Laparoscopic Surgery  
LISA MÅNSSON

© LISA MÅNSSON, 2020.

Supervisor: Klas Modin, Torbjörn Lundh, Department of Mathematical Sciences  
Examiner: Torbjörn Lundh, Department of Mathematical Sciences

Master's Thesis 2020  
Department of Mathematical Sciences  
Division of Applied Mathematics and Mathematical Statistics  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: Illustrative summary of how to use landmarks, a camera model, and camera calibration to obtain a tumour projection for intraoperative guidance.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Gothenburg, Sweden 2020

# **A Model for an Augmented Reality Tool in Tumour Removal Laparoscopic Surgery**

LISA MÅNSSON

Department of Mathematical Sciences

Chalmers University of Technology

## **Abstract**

One of the most lethal types of cancer is hepatocellular carcinoma, cancer in the liver. Because of the many risks entailed with open surgery, the use of laparoscopic surgery has increased, with no exception in liver resections. Instead of a big cut, minimally invasive techniques are used, placing small ports on the abdomen where surgical tools as well as a laparoscope can be inserted. The surgeons orient themselves from the outside, creating a perception of the inside through the 2D images from the camera and a preoperative 3D image on the side as a map. Since the liver is an essentially homogeneous organ, it can be hard to orient from this information, why surgeons in Gothenburg have developed markers to place on the liver's surface. With help of such markers, the goal is to develop an augmented reality tool for intraoperative guidance, mapping the laparoscopic 2D image to the corresponding 3D position, and be able to project a tumour area in the laparoscopic view. In this work, an inventory of laparoscopic liver resection was performed and a camera model as well as a simulated liver environment was developed. To map a 2D image to the 3D environment, an algorithm for estimation of the camera pose, named POSIT, was examined. It was concluded that the limitations of POSIT were not compatible with the problem, but the platform developed, consisting of a simulated liver environment and the camera model, can be used as a framework for future work, in which testing of other algorithms to estimate the camera pose can be performed.

**Keywords:** Augmented reality, camera model, camera calibration, POSIT, laparoscopic surgery, minimally invasive surgery, cancer, liver resection, hepatocellular carcinoma.



# Acknowledgements

Firstly, I would like to acknowledge Torbjörn Lundh and Klas Modin for all the time you spent with me this autumn, discussing the project, and supporting me. Accessing two such diverse supervisors in my master's thesis work has been very beneficial, and the all-time positive attitude to the project that you both have in common is highly appreciated. All meetings with you have lead to valuable insights and conclusions – both with and without Torbjörns' office candles.

Niclas and Mårten, it has been a pleasure to collaborate with you. I'm sincerely grateful for you and your colleagues' hospitality, letting me take part in surgeries as well as your ongoing research, for you being curious and involved in my work, and for answering all my questions. It has been eye-opening to work with people from multiple disciplines, highlighting the importance of continuous conversation with the beneficiaries of a solution. From explaining and learning in both directions, we have all seen that great ideas can grow.

Thanks to Siemens, I also got the opportunity to attend a surgery at Oslo university hospital. Again, every person I met gladly answered my questions. Also, without the helpful explanations from Maria at Olympus my understanding of the laparoscope would have taken at least twice the time.

As this is the end of a five year long adventure at Chalmers University of Technology, I would like to declare a special thank you to my family who always persisted physics talk and took me down on earth during stressful periods, and another to my friends in Engineering Physics and my master's without whom I would never have gotten the same results, experiences and insights, or be where I am today.

Lisa Månsson, Gothenburg, January 2020





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Previous work . . . . .	2
1.1.1	AR with intraoperative ultrasound imaging . . . . .	2
1.1.2	AR based on a static 3D image . . . . .	2
1.1.3	Registration difficulties . . . . .	3
1.1.4	ICG immunofluorescence . . . . .	3
1.2	Aim . . . . .	4
1.3	Issue specification . . . . .	4
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	The forward model – from 3D to 2D . . . . .	7
2.1.1	Camera model . . . . .	7
2.1.2	Camera extrinsics, translation and rotation . . . . .	8
2.1.3	Pinhole camera projection . . . . .	9
2.2	The inverse problem and camera calibration . . . . .	10
2.2.1	Algorithms and existing code . . . . .	11
<b>3</b>	<b>Methods</b>	<b>13</b>
3.1	Inventory of laparoscopic liver resection . . . . .	13
3.1.1	Camera specifications . . . . .	13
3.1.2	Dimensions and cases of interest . . . . .	15
3.2	Simulation of data . . . . .	16
3.2.1	Camera paths . . . . .	17
3.2.2	Summary . . . . .	17
3.3	Proposed inverse model – POSIT . . . . .	17
3.3.1	General concept . . . . .	18
3.3.2	Modern POSIT . . . . .	20
3.3.2.1	Homogeneous coordinates . . . . .	20
3.3.2.2	Mathematical description . . . . .	20
3.3.3	Advantages and performance . . . . .	21
3.4	Evaluation of a found pose – error estimation . . . . .	22
<b>4</b>	<b>Results</b>	<b>23</b>
4.1	Forward model validation . . . . .	23
4.2	Inverse model - evaluation of POSIT . . . . .	24
4.2.1	Static example . . . . .	25

4.2.2	Translational movement . . . . .	26
4.2.2.1	Fixed orientation . . . . .	26
4.2.2.2	Non-fixed orientation . . . . .	28
4.2.3	Circular movement . . . . .	30
4.2.4	Environment variation . . . . .	32
4.2.5	Conclusions from POSIT evaluation . . . . .	33
<b>5</b>	<b>Discussion</b>	<b>35</b>
5.1	Future work . . . . .	36
<b>6</b>	<b>Conclusion</b>	<b>37</b>
	<b>References</b>	<b>39</b>
<b>A</b>	<b>Appendix</b>	<b>I</b>
A.1	Issues in dynamic soft tissue tracking . . . . .	I

# 1

## Introduction

Cancer, one of today's most threatening diseases worldwide, exists in more than a hundred different types. The third most lethal is called hepatocellular carcinoma, HCC, and is the most common type of liver cancer. It takes about 700 000 lives every year [1]. Nevertheless, no proper antidote has yet been found and sick tissue has to be resected in the purpose of trying to get rid of the disease [2]. Ablation through open surgery, entails high risks of, for example, infections and postoperative complications. Laparoscopic surgery is a technique where minimally invasive methods are used. Instead of a big incision in the patient's belly, small holes called ports are made for insertion of tools fastened on sticks, as well as a camera-stick called a laparoscope, giving a 2D view of the operation scene from the inside.

Laparoscopic liver surgery has many advantages such as shorter hospitalization, lower morbidity and less postoperative complications. Small incisions avoid disruption of the abdominal wall of muscles and less adhesion formation makes future surgery easier [2, 3]. Furthermore, a smaller amount of positive resection margins has been seen, meaning less cases where not all of the malignant tissue was removed, in comparison to open surgery. It is of high importance to resect the whole tumour so that no cancer cells are left in the body. Although, the more tissue removed, the higher risk for postoperative liver failure (with a mortality around 30%). Therefore, it is crucial to have high precision in liver tumour resection surgery, to not take too little, nor too much [2].

Although laparoscopic surgery has many advantages, it is a difficult technique to learn that needs a lot of practice. It is especially complicated with liver tumours, since the liver is an unusually homogeneous organ, implying that it is very hard for the surgeons to orient themselves in the 2D camera view. Furthermore, in open surgery, the liver surgeon palpates the liver to feel the tumour, when in the case of laparoscopic surgery haptic feedback is missing. To their help they can make a three dimensional image<sup>1</sup> of the area of interest, where the tumour can be visible, to use as a map. Although, the tumour is in most cases not visible on the live 2D image from the laparoscopic camera and this, in addition to the fact that the liver has a very homogeneous structure, makes it hard for surgeons to orient, even with a 3D image on the side. A lot of experience is needed to map the pre-generated 3D view to the live view in 2D. The surgeon needs to look from one screen to another and compare the two throughout the whole surgery.

---

<sup>1</sup>For example a CT-scan (computed tomography) or dyna-CT scan, made with X-rays.

The problem stated in the previous paragraphs is a known issue in the medical field, and there have been different research projects going on for improved precision in laparoscopic surgery previously. In the following section, a few examples will be mentioned, including limitations and advantages.

### 1.1 Previous work

The definition of an Augmented Reality (AR) is an artificial environment where information is added to the real world [4]. The most famous example is probably the game “Pokémon Go”, where you walk around in your actual environment, and on the screen sees the same scene, but with Pokémons around you. Virtual reality (VR), on the other hand, is not showing images from the real environment, but builds up a world on its own. Both AR and VR have been seen very useful for surgery planning as well as good tools for surgery training where 3D models of the liver are available, environments comparable to the real operation space [5]. AR can also be beneficial in planning the positioning of laparoscopic ports [6].

In liver surgery, the most important factor for success is the ability of visualizing intrahepatic structures<sup>2</sup>, to know the tumour’s position itself but also in relation to blood vessels and biliary ducts to not cause damage. Recent research also focuses in taking advantage of AR intraoperatively, thus intrahepatic structures could be displayed live, during surgery [7].

#### 1.1.1 AR with intraoperative ultrasound imaging

A common helping tool in the operating room is intraoperative ultrasound, although, it is not always an easy task for the surgeon to orient in, interpret, and map these 2D images to the liver’s 3D anatomy [2]. Lung et al. (2019) came up with a solution where the image of a laparoscopic ultrasound (LUS) was fused with the laparoscopic camera view. The LUS stick was visible in the camera view, facilitating the navigation. An electromagnetic tracking system was successfully used in the registration process, allowing for a precise placement of the projected LUS image in the laparoscope’s view, and no need for preoperative imaging for the registration<sup>3</sup> [8]. A drawback with this method is that it requires an extra port, and it is hard to operate with the surgical tools as long as the LUS is used, because it is in their way.

#### 1.1.2 AR based on a static 3D image

Hallet et al. (2015) succeeded to perform an AR assisted trans-thoracic minimally invasive liver resection. A preoperative 3D CT image was mapped to the real time 2D laparoscopic camera view. The process was manually performed, merging of

---

<sup>2</sup>The different structures inside the liver.

<sup>3</sup>Registration, the process of aligning the virtual world with the real world.

the laparoscopic image and the preoperative 3D CT-image by localization of visible landmarks. Since the 3D image was static, movements or changes in the tissue made it outdated. Furthermore, the mapping process needed a computer scientist present and was not automatized [9]. The general idea of this technique is very similar to the goal of this project, but implemented in a less automatized way. If imaging was done intraoperatively, there might have been a better correlation between the 3D image and the live 2D view considering a dynamic case, especially if there was a possibility to update intraoperative 3D images. Although, there are still some main drawbacks with such a solution: it is both difficult and expensive to integrate an advanced imaging system in an already crowded operation room [10, p. 625].

### 1.1.3 Registration difficulties

One of the most challenging steps when involving AR in intraoperative processes is the registration, the mapping between the real world and the positions of the extra information that is to be added. This process requires some kind of trackable landmarks, either anatomical or inserted ones, e.g. fiducials, a type of marker that is glued to the skin of patients, or bolt heads screwed into the bone [10, p. 630]. In one study, minimally invasive tattooing on the liver (through one port in the belly) was performed preoperatively under guidance of ultrasound to mark out the tumour area [11].

### 1.1.4 ICG immunofluorescence

There are also solutions proposed without incorporation of any AR or VR systems. By intravenously injecting a fluorophore called Indocyanine green (ICG), previously used to predict liver failure, the substance metabolizes in the liver cells [12]. Cheung et. al. (2018) used ICG immunofluorescence in laparoscopic liver surgery and had positive outcomes and no drawbacks seen in comparison to the open surgery option. With this method it is not only possible to track the targeted tumour, but also find additional lesions that were not discovered preoperatively and might be necessary to remove. There is also a better identification of the bile duct structure [13]. Although, there is a risk of false positives that might need to be verified with another method [5]. An option might be photo-acoustic imaging, a combination of optical and ultrasound imaging, and a new non-invasive method for medical imaging giving a very high image resolution and large penetration depth. The tumor boundaries can be determined in the molecular scale and the method is promising in both diagnosis and intraoperative guidance, perhaps in combination with fluorescence navigation [5]. Even if this might be a good solution, the fact remains that the penetration depth of IR light is just about 10 mm, implying that tumours located deeper down in the liver won't be emitting the fluorescent green light. Another drawback with ICG is that the method requires a special laparoscopic imaging system to see the fluorescence [13].

### 1.2 Aim

There is currently no existing solution to the intraoperative navigation problem independent of extra equipment in addition to the existing one in an operating room. We want to examine if there is a possibility to keep track of intrahepatic structures by the use of only the laparoscope and a known 3D structure relating the tumour's position to landmarks on the liver.

Based on a surgeon's difficult task to, by experience, manually and imaginary merge a laparoscopic 2D view with a preoperative 3D image, this work aims to facilitate it with an Augmented Reality tool, projecting the tumour area to its position on the 2D screen. Surgeons at Sahlgrenska University hospital in Gothenburg, under the direction of Mårten Falkenberg and Niclas Kvarnström, have improved the orientation during surgeries by preoperatively inserting a few small landmarks, visible in the 3D image as well as in the 2D laparoscopic live view. Our hypothesis is that by knowing the position of these landmarks and their relative position to the tumour, the 3D view can be projected down to two dimensions and the useful information can be mapped to the laparoscopic image and projected, creating an augmented reality. The hope is that with such guidance increase accuracy in liver resections through a solution where no additional equipment is added to the already crowded operation room.

AR applied to abdominal surgery is a greater challenge than other types of surgery, because of a significant amount of organ movement and deformation [6]. In this thesis, the liver will be considered a solid body, with a geometry as in a preoperative 3D image, as a first stage. The dynamic problem is therefore further ignored, but will be briefly discussed in Appendix A.

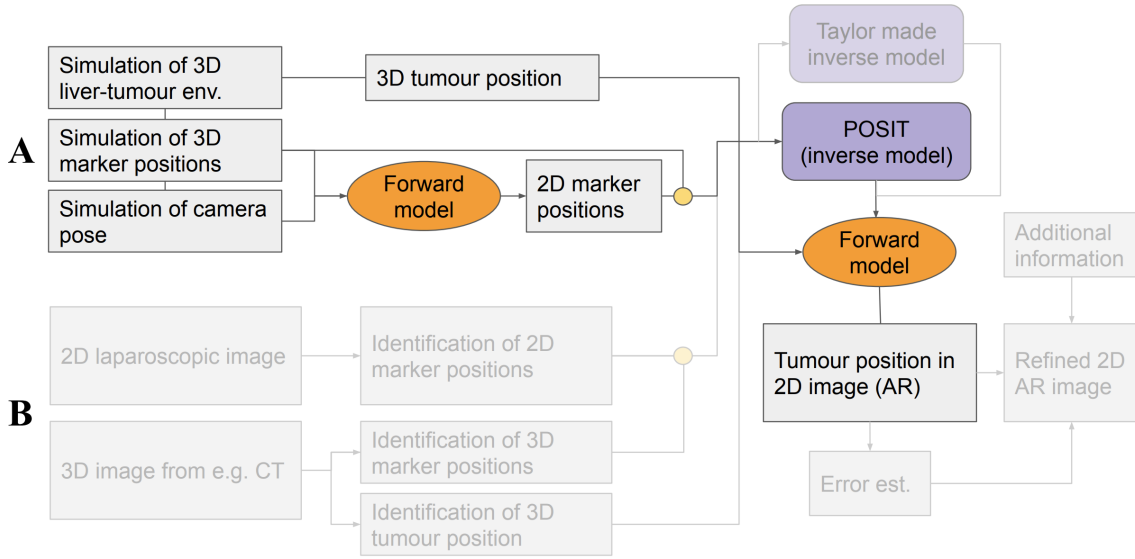
### 1.3 Issue specification

This thesis could be divided into three specified sub-tasks:

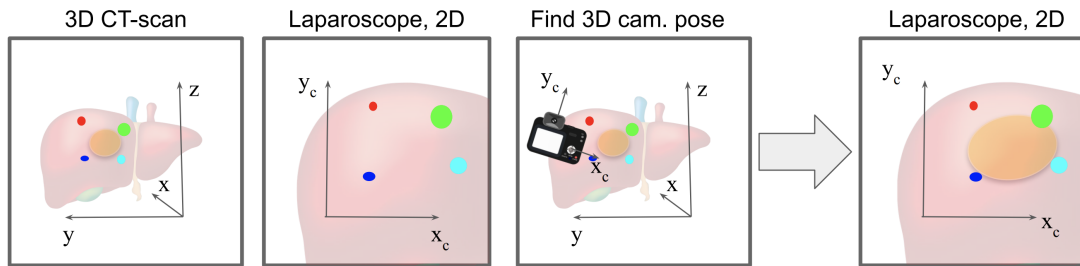
1. Inventory of a laparoscopic liver resection, to better understand possibilities and limitations.
2. Adapting a suitable camera model to the laparoscopic system, and simulate projections from a 3D liver environment to the 2D image plane of the camera.
3. Finding a suitable camera calibration algorithm based on the parameters in the camera model – a solution to the inverse problem of knowing 2D projected positions of markers on the liver, and map these to a preoperative 3D image. Knowing the camera's position in 3D it is possible to project the tumour onto the laparoscopic image.

An overview of the proposed solution to the problem stated in section 1.2, as a whole, can be seen in Figure 1.1. The different blocks will be further explained in the upcoming sections. This thesis work is only considering a part of that full solution, illustrated separately in Figure 1.2. As an example, the future input to a

full solution are real laparoscopic images with some kind of feature points as well as a 3D image where positions of tumour and feature points have to be defined (path B in Figure 1.1). The gap between that reality and this work is an algorithm to find positions of feature points in the image as well as compatibility with a software where the points in 3D can be defined – either manually or by a machine. After tumour projection there will also be a need for uncertainty estimations, improvements, and design of an interface for an AR implementation, updating the laparoscopic image with tumour position (perhaps other data too) in real time.



**Figure 1.1:** An overview of the proposed solution as a whole: a non-invasive augmented reality tool for increased precision in tumour removal laparoscopic surgery. The faded blocks are not treated in this work. Path A is for simulating test data, and path B describes the steps when working with real data from laparoscopic surgery.



**Figure 1.2:** An overview of the work in this thesis, corresponding to the non-faded blocks in Figure 1.1, following path A. Positions in 3D for liver, tumour (in brown) and markers, as well as a certain camera pose (position and orientation), are simulated. With a camera model (forward model), the 2D projected points of the markers are obtained. Based on the simulated data from markers' positions in 3D and 2D, the camera pose can be estimated (and compared with the true, simulated one) and thus the tumor position projected onto the 2D laparoscopic image (AR).

## 1. Introduction

---

For the specified task, one needs some background information about mathematical tools such as geometric projections, camera models, optics and calibration, discussed in the upcoming sections.



# 2

## Background

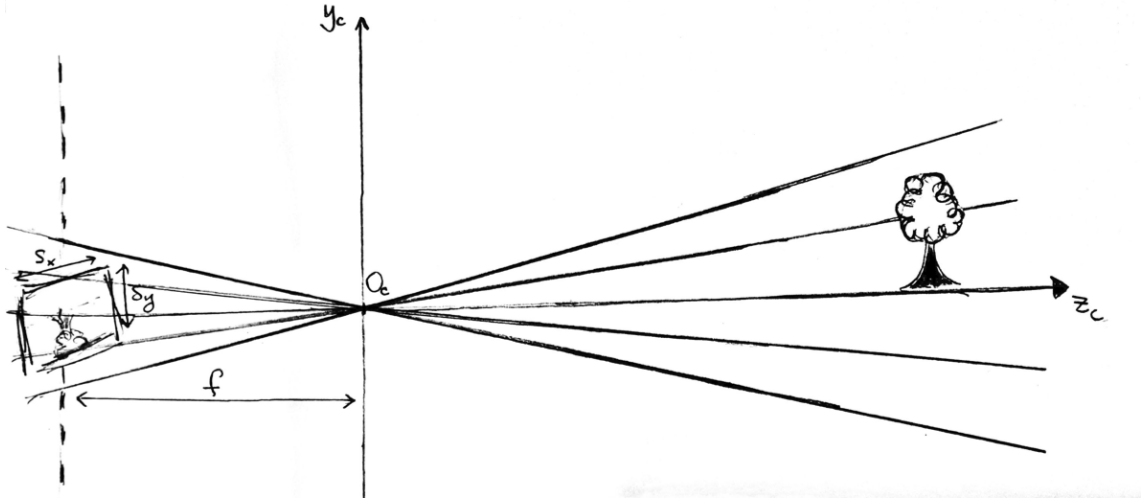
In this section, the theory behind development of a *forward camera model* is presented, an algorithm using 3D information to create a camera image in 2D. This is an important part of the project, not only in simulating test scenarios, but also in the projection of a tumour on a 2D image by knowing its 3D position and the camera's pose. Secondly, the inverse problem is introduced: how to determine the 3D position of an object from a 2D image. While reading chapter 2 and 3, it might be a good idea to keep Figure 1.1 and 1.2 in mind, to better understand where the different parts explained are used.

### 2.1 The forward model – from 3D to 2D

To model a camera in a representative way, it has to be understood optically. There are different kinds of projections when moving from three dimensions down to two. Orthographic projection, also called orthogonal projection, is as the name tells the 2D projection on a plane (the image plane) where all projection lines are perpendicular to this plane, implying that information about depth will get lost in the projection. Perspective projection, on the other hand, takes depth into account, and makes for example parallel lines converge further away. The points in 3D are projected to 2D along a set of projection lines that meet at one certain point, called the center of projection [14, p. 407].

#### 2.1.1 Camera model

The simplest model of a camera is the so called pinhole model, consisting of a parallel projection with a 180 degree rotation of the image, illustrated in Figure 2.1. It is basically a model of a box which has a small pinhole in one of its sides. This is where the light enters and an upside-down image of what is outside the box is projected on the opposite wall, inside the box. The model has a simple geometric setup, but can be extended. For example, a pinhole does not gather enough light for rapid exposure and for that reason our eyes as well as today's cameras have lenses included in the system. A lens can gather more light, but it also causes distortions to the image that has to be compensated for in a camera model for such a system [14, p. 370]. Neglecting a lens system, the focal length together with the image size determine how the projection will be, in terms of field of view and sizing and positioning of different objects, visualized in Figure 2.1.



**Figure 2.1:** Illustration of the pinhole camera model. The parameters shown in the illustration are:  $f$ , the focal length,  $S_x$  and  $S_y$  setting the size of the image plane, and  $O_c$ , the center of projection – and the pinhole in the box. The up-side-down image to the left is supposed to be on the opposite wall of the pinhole camera box. The  $z_c$  axis shows the viewing direction of the camera, and the  $y_c$  represents its up-direction.

### 2.1.2 Camera extrinsics, translation and rotation

The extrinsic parameters of a camera are the camera position ( $O_{xc}$ ,  $O_{yc}$ ,  $O_{zc}$ ) together with its rotational orientation that can be described in various ways. We will now consider two coordinate systems, one called the world coordinate system and the other called the camera's coordinate system. The last one is fixed to the camera and follows its translations and rotational movements. The camera's coordinate system is oriented such that the positive  $z_c$  axis points in the direction of the camera view, and the positive  $y_c$  axis points up from the top of the camera, sometimes called the up-vector, visualized in Figure 2.1. The direction of the  $x_c$  axis follows from the cross product of  $y_c$  and  $z_c$ , as defined in a right-handed coordinate system.

In this forward model, the points in space are in the first place defined in the world coordinate system, with an arbitrary origin. The camera position can then be expressed in this system. An orientation of one coordinate system relative to another can be described in many ways, but here we will use Euler angles for rotation in three dimensions. Starting from the world coordinate system axes,  $x$ ,  $y$  and  $z$ , we can choose three Euler angles for rotation around each axis to describe the camera's orientation.

A rotation<sup>1</sup> of a vector with an angle  $a$  around the  $x$ -axis can be described by the rotation matrix

$$R_x(a) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix} \quad (2.1)$$

For rotations around  $y$ - and  $z$ -axes the matrices look similar, as can be seen in Equation (2.2) and (2.3). Here, we rotate around  $y$  with an angle  $b$ , and around  $z$  with an angle  $c$ .

$$R_y(b) = \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix} \quad (2.2)$$

$$R_z(c) = \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Multiplying matrices  $R_x$ ,  $R_y$  and  $R_z$  a matrix for the full 3D-rotation is obtained,  $R = R_x(a)R_y(b)R_z(c)$ . Transposing any of the matrices  $R_x$ ,  $R_y$ ,  $R_z$  or  $R$  will instead of rotating a point rotate the coordinate system<sup>2</sup> in which the point is expressed in<sup>3</sup>. This means that the resulting vector from the multiplication will be the vector of the same fixed point in space but expressed in a rotated coordinate system.

With the matrix  $R$ , we obtain the combination of three upon each other following rotations starting with  $z$ , then  $y$  and lastly  $x$ , with a direction of rotation following the right-hand rule<sup>4</sup>, applied to a point in the world coordinate system. Apart from separating active from passive rotations, it is also important to know the difference between extrinsic and intrinsic rotations. The Euler angle matrices used are defined as above for *extrinsic* rotations, thus the second rotation will be with an angle  $b$  around the world coordinate system's  $y$ -axis. Continuously, the last rotation to reach the camera's orientation is done with the angle  $a$  around the world coordinate system's  $x$ -axis. The opposite, intrinsic rotations, would instead have used the new  $y$ -axis,  $y'$ , obtained from the first rotation in  $z$ , for the second angle  $b$ . Similarly, the last intrinsic rotation with the angle  $a$  would have been around the  $x'$ -axis, obtained from the two previous rotations with angle  $b$  and  $c$ .

### 2.1.3 Pinhole camera projection

When points of an object undergo translation and rotation according to the previous section, the result is expressed in the camera's coordinate system,  $x_c$ ,  $y_c$ ,  $z_c$ , instead

<sup>1</sup>An active rotation, when a point is moved around the coordinate system.

<sup>2</sup>The opposite of an active rotation, called a passive rotation.

<sup>3</sup>To not be confused: The passive rotation matrix is the default matrix generated by the Matlab function `angle2dcm.m` (<https://se.mathworks.com/help/aerotbx/ug/angle2dcm.html>). The angles on the input to this function can be multiplied by -1, thus giving the transpose of the matrix due to its symmetric components.

<sup>4</sup>When the right-hand's thumb is in the direction of the axis, the fingers are curling in the rotational direction.

of in the world system. From here, the points will undergo the pinhole camera projection. This corresponds to a projection based on the focal length  $f^5$ , as well as the intersection of the camera's optical axis with the image plane, called the principal point  $(c_x, c_y)^6$ . These parameters are some of the intrinsic ones of the camera, and together they can form the so called camera matrix, shown in Equation (2.4). Dividing all components of a vector with its  $z$ -component<sup>7</sup>, followed by a multiplication with this matrix, will project the points from 3D down to an image plane in 2D. Performing this transformation, objects further away will shrink, as well as appear closer to the optical axis, than closer objects with the same distance to it will do. A perspective projection is obtained [14].

$$C = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

By performing this transformation it is possible that two or more points will get the same  $x, y$ -coordinates in the 2D image plane. To know which point lays closer to the observer than another, a sorting algorithm is implemented. This sorts all observed points in the 3D space after their distance<sup>8</sup> to the camera and makes it possible to know in which order to plot them: the points furthest away first, to then overwrite these with the closer ones, if any.

## 2.2 The inverse problem and camera calibration

Seeing multiple 2D views of a 3D object from different angles gives a feeling for what the 3D structure looks like in the real world. Knowing the precise positions of, and distances between, some points on that 3D object, the camera pose, meaning its position and orientation, does not have to be known to map the projected 2D image points to the 3D points in the real world [14, p. 378], but can be estimated. There are also other methods for this, that does not require more than a single image to specify positions.

With a camera calibration, a 3D world can be reconstructed from 2D images. The purpose of a calibration is to find the camera parameters, usually divided into two subgroups: intrinsic and extrinsic parameters, discussed briefly in the previous sections. The intrinsic parameters are the ones depending on how the camera is built up, such as radial and tangential lens distortion parameters, focal length, and the principal point of the camera. The extrinsic parameters describe the camera's pose: position and orientation. The position is defined by the  $x$ -,  $y$ - and  $z$ -coordinates relative to a world coordinate system. The orientation is described by, for example, a rotation matrix based on three Euler angles, previously discussed. Figure 2.2 gives an illustration of the transformation from one coordinate system to another.

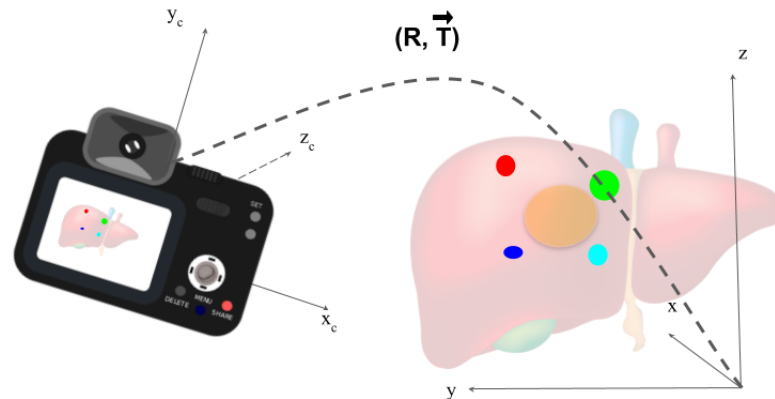
---

<sup>5</sup>Here, squared pixels are assumed. If this is not the case, two different focal lengths,  $f_x$  and  $f_y$ , are used [14, p. 373].

<sup>6</sup>Here, the principal point of the camera is assumed to lay in the center of the image plane.

<sup>7</sup>Basically the concept of homogeneous coordinates, discussed in section 3.3.2.1.

<sup>8</sup>In fact, the squared distance,  $x^2 + y^2 + z^2$ , avoiding computations of roots.



**Figure 2.2:** Image illustrating the transformation (rotation  $\mathbf{R}$  and translation  $\vec{\mathbf{T}}$ ) from a world coordinate system to a system fixed to the camera.

### 2.2.1 Algorithms and existing code

Algorithms for camera calibration exist in Matlab<sup>9</sup> as well as in other programming languages [15]. Most options seen are based on that a specific two-dimensional calibration pattern, in most cases a checkerboard, is present in the position where to perform projections from 3D to 2D. In this case, such algorithms were rejected, since the positions of feature points on the liver's surface most likely won't be in the same plane, neither in a squared pattern.

An algorithm named POSIT requires the intrinsic camera parameters as an input, but can calibrate for the extrinsic ones independently, without the need of a coplanar construction of points. In addition to the intrinsic parameters, the algorithm needs the dimensions of the 3D object to be projected, based on at least four non-coplanar points on its surface.

<sup>9</sup>Matlab also has an app with a GUI, where basically no programming skills are needed.



# 3

## Methods

In this section, various steps performed to reach the final conclusion of this thesis are presented. Firstly, a more specified description of the problem is presented in terms of liver anatomy and laparoscope properties, based on surgical experience. With these findings, a simulation of a liver environment as well as realistic test cases could be developed. The algorithm called POSIT is proposed for solving the inverse problem using a 2D image to estimate the camera pose, and will be further explained. Lastly, a suggested error estimation of camera poses is presented, useful when the true camera pose is known. All code and algorithms presented in this thesis were implemented in Matlab 2018b.

### 3.1 Inventory of laparoscopic liver resection

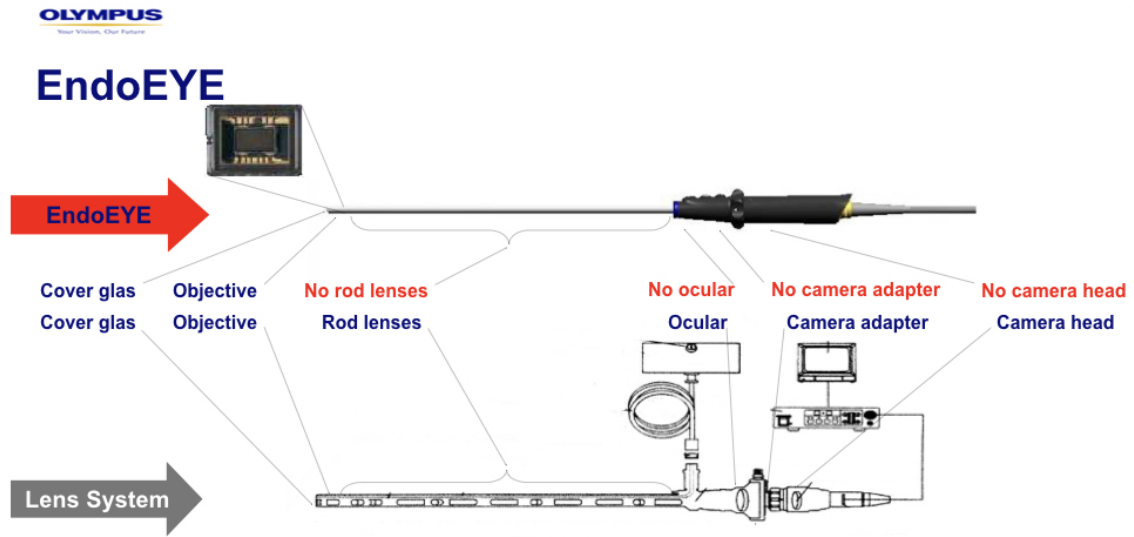
Attending two laparoscopic liver resections, a more specified description of the problem was obtained and further developed through information exchange with surgical staff at the hospital and their cooperators. The specifications are presented in terms of laparoscope properties and liver anatomy below.

#### 3.1.1 Camera specifications

The current laparoscopes used by the liver surgeons at Sahlgrenska University hospital are from the manufacturer Olympus. There are two different camera set-ups: a video laparoscope, or, a camera head plus a stick connected to it with the optics. Figure 3.1 shows an illustration of the two different setups. A camera head contains the CCD<sup>1</sup> chip where the video is recorded. This part of the laparoscope is always outside the body, and it is only the stick entering the laparoscopic port. To enable the light to reach the chip in the camera head, there are optical components, lenses, inside the stick. In a video laparoscope, there are very few lenses inside the stick because the CCD chip is located at the end of it, just where the light enters. This implies that the image from a video laparoscope has better color, contrasts, and is less noisy, than an image from a system with more lenses.

---

<sup>1</sup>Charged Coupled Device, a sensor converting light energy to an electric signal.



**Figure 3.1:** Illustration showing the difference between a video laparoscope (here, Olympus’ EndoEye) and a setup with camera head and connected optics. Image from Maria Karltenius, Olympus. Reproduced with permission.

There are mainly two different laparoscopes used in liver surgery at Sahlgrenska in Gothenburg. One is the EndoEye 3D from Olympus, a video laparoscope with two CCD chips in the end of the stick, providing a (small) depth in the image when 3D glasses are worn looking at the live image in the operation room. The other laparoscope is based on the Olympus camera head named CH-S200-XZ-EA. In both models there is an auto focus function, and the focal length of the CH-S200-XZ-EA camera head is given as an interval: 15-30 mm. For the EndoEye there is no focal length specified<sup>2</sup>. Using that the diameter of the laparoscope tip is 10 mm and assuming a chip resolution in the order of  $1000 \times 1000$  pixels, a focal length of 15-30 mm corresponds to 1500-3000 pixels. Thus, a focal length of 3000 pixels was used in the simulations. Furthermore, attending a surgery performed with EndoEye 3D, there were no lens distortions visible in the live image<sup>3</sup>. Because of this, the simulations were performed using zero-values for all distortion parameters in the camera model.

A suitable forward model, adaptive to the laparoscopic cameras specified, was developed in Matlab by Zachary Taylor in 2014 [16]. In this model, the focal length of the camera is needed as an input. In the laparoscopes used, the focal length is not kept constant, nor shown or given by the system at any time. Since this parameter is required, this will be an issue when working with real laparoscopic images. It is possible that a value within a certain range is accurate enough. If not, one solution might be to manipulate the laparoscope to have a fixed focal length. Another solution could be to keep track of the focal length, but the manufacturer does not seem to provide such data.

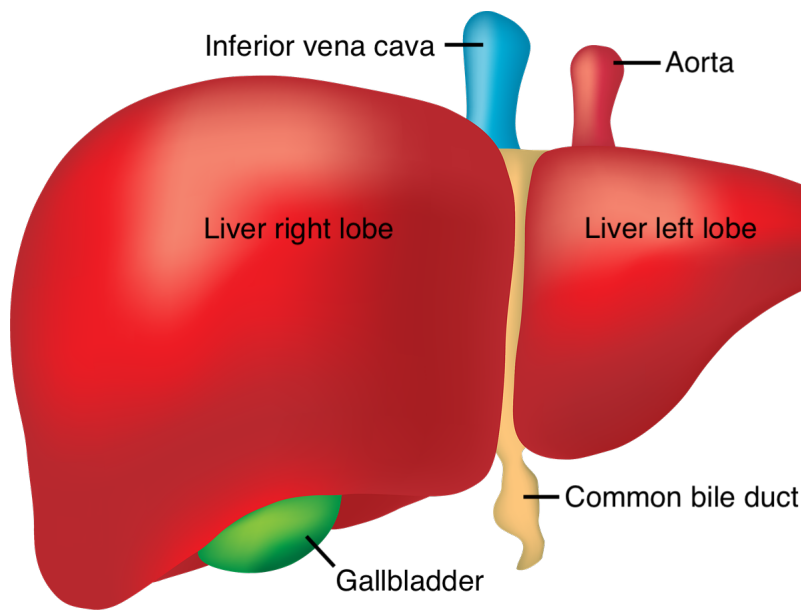
<sup>2</sup>These values are hard to get specified.

<sup>3</sup>The camera head CH-S200-XZ-EA was not used in the surgeries during this inventory.



### 3.1.2 Dimensions and cases of interest

In a liver tumour resection, the resection margins lay in between 5-10 mm. This is specifying how much of the tissue around the tumour area that has to be resected. It is common that the resected volume is similar to a frustum (cut off cone) or a semi sphere, like it is “scooped out”. The reason for this is that the tissue between the tumour and the liver’s surface cannot be saved anyway, because of disconnected blood supply. All cells in the liver are connected to the blood circulatory system through capillaries with blood supply from only one entrance in the organ.



**Figure 3.2:** Illustration of a human liver.

Different tumours in the liver are more or less easy to resect. Cases that do not have a need for an augmented reality solution are for example: when a small tumour is located just by the liver’s surface, when a tumour is located in the “tip” of the liver (the more narrow part in the left lobe, see Figure 3.2) – implying that the tip can easily be cut off, or, when a small tumour is located very deep, and a “boiling” technique is used – not to be applied to tumours bigger than 2 cm in diameter. The typical case where AR guidance is needed is a tumour located in the “bulk” part of the liver, where the surface has a radius of curvature of about 10-15 cm. The tumour area seen from above the liver surface is typically in the range of 3-10 cm, with a tumour positioned at a depth of 1-3 cm reaching 3-7 cm at its end. If the tumour is located deeper than that, it is positioned close to bigger blood vessels, implying that the whole vessel has to be resected, meaning a greater part of the liver. Thus AR guidance is again not needed. Some tumours are easily discovered and visible on the surface of the liver, but still a case of interest because of the missing depth information. The above stated numbers were used in order to create the table of specifications, Table 3.1, and to further on simulate a realistic liver environment.

**Table 3.1:** Table with parameters for setting up a realistic environment of a liver, tumour and markers on the liver surface. The liver’s and tumour’s principal radii of curvature are to determine suitable radii of spheres approximating the liver and tumour volumes respectively.

Parameter	Designation	Range [mm]
Radius of curvature, critical liver surface	$\rho$	100-150
Tumour min depth	$d_{min}$	10-30
Tumour max depth	$d_{max}$	30-70
Max diameter of tumour seen from surface	$l$	30-100
Camera working distance from liver	-	30-100
Radius of curvature, tumour surface	$r$	$\max(d_{max} - d_{min}, 1)$

For an AR projected tumour to be valuable in liver resections, the precision desired is estimated to be a max error in 3D within 2 mm. This error arises from two or more sources: the algorithm used in this solution, and the registration error from the process of selecting the tumour volume in a 3D image.

## 3.2 Simulation of data

To go backwards, discover a 3D environment from 2D projections, the algorithm POSIT performs a camera calibration to find the extrinsic parameters. This means that it computes an approximate pose of the camera, consisting of a translation vector (its position) and a rotation matrix defining its orientation. To perform a camera calibration, the algorithm needs simulated data in terms of 3D feature points and their corresponding 2D projections, since real data from operations was not yet an option because of the lack of an algorithm for marker detection.

By setting up a 3D environment with markers and any other objects in the 3D world, these can be projected down to a two dimensional image plane, using the forward camera model described in section 2.1. To do this, some intrinsic parameters of the camera are needed, as well as the extrinsics, defined by a camera position,  $O_C$ , and an orientation, described by three Euler angles, here represented by a rotation matrix.

In this work, testing environments<sup>4</sup> based on a liver environment as realistic as possible were built, including two spherical representations of the liver and tumour respectively, as well as single points in different colours<sup>5</sup> placed on the liver’s surface, representing the markers. The simulated liver-tumour environment was based on parameters in the ranges presented in Table 3.1, and from this, the markers’ configurations were varied, creating environments with different distances between markers (M-M) and number of markers. Nine environments were examined, presented in Table 3.2, and graphically presented in terms of projected markers seen

---

<sup>4</sup>By *testing environment* means the number of markers and their positions in 3D.

<sup>5</sup>Working under the assumption that the markers are distinguishable.

from the camera in Figure 4.14. Other environments can easily be added to the code in future work.

**Table 3.2:** Environments tested for set up of simulations. Mean distance M-M stands for mean distance in between the markers.

Environment	Mean distance M-M [mm]	Number of markers
A	4.0	4
B	4.8	4
C	5.6	4
D	7.0	4
E	8.1	4
F	9.5	4
G	32.8	4
H	42.8	4
I	45.9	10

### 3.2.1 Camera paths

The environments were examined in static scenarios, but some of them followed upon each other in a camera path. Two paths were simulated: one straight line, where the camera moves further away from the liver, and one circular path, where the camera is kept at the same  $z$ -height in the world coordinate system<sup>6</sup>.

### 3.2.2 Summary

The different parts presented in this section – setting up a liver-tumour environment, simulating and characterizing marker configurations as well as simulating camera paths – shall be seen as a framework in which different algorithms/code for both the forward and inverse model can be tested. Perhaps smaller modifications in input data will be needed depending on the algorithms to test, but it is supposed to be a universal framework, where new test cases, both environments and camera poses and paths, can be added. Apart from setting up a test case, there is also code for visualizing performance in a test, and a function for error estimation of results from an inverse algorithm’s camera pose estimation when the true pose is known, the last one further described in section 3.4.

## 3.3 Proposed inverse model – POSIT

In 1995 Daniel DeMenthon and Larry S. Davis wrote the paper *Model-based object pose in 25 lines of code* [17], presenting an algorithm for pose estimation based on

<sup>6</sup>To create smooth paths, like the circular one, a function called `cscvn` (<https://se.mathworks.com/help/curvefit/cscvn.html>), from Matlab’s curve fitting toolbox, can be used, with no need for conversions to spherical coordinates, simply by defining points through which the camera shall pass.

a single image with markers/feature points, and a known structure of the image object carrying those features points, called POSIT. The algorithm assumes that the distance between the object with feature points and the camera ( $Z_0$  in Figure 3.3) is greater than the distance in between the feature points themselves. Another important presumption for the algorithm to work is to have at least four non-coplanar object feature points. POSIT has been used to find poses in many areas of research since 1995, tracking poses from various things from faces to spacecrafts. The algorithm is briefly explained below and a more detailed description can be found in the original paper [17]. The code can be found on the web page referred to in Reference [18].

#### 3.3.1 General concept

A camera's pose consists of its position vector as well as its orientation defined by a rotation matrix, both in relation to some arbitrary coordinate system, here called the world coordinate system. POSIT is in fact a combination of two algorithms. POS (Pose from Orthography and Scaling) uses a scaled orthographic projection as an approximation for the actual perspective projection, to be able to solve for the camera's pose by solving a simple linear system. The last two letters in the abbreviation stands for *iteration*, in total POS with ITeRations. POSIT uses an iteration loop to converge the approximated solution from POS towards the true camera pose. In each iteration, the pose just found is used together with the object's coordinates to update the projected feature points<sup>7</sup> (the markers), in order to get the positions of the (perspective-)projected feature points closer to scaled orthographic projections. In this way, the pose solution from POS will get closer to the real camera pose.

By simultaneously looking at and referring to the vectors in Figure 3.3, the understanding of the previous description of the algorithm may be facilitated here. The object's world coordinate feature points are the points named  $M_i$  (only one such point is shown in the figure). The perspective projection of  $M_i$  lands in  $m_i$ , in the image plane  $G$ , at a distance of the focal length  $f$  from the camera. The image coordinates  $m_i$  are known as the perspective projections of the feature points onto the image plane, and are given as an input to the algorithm. The scaled orthographic projection (SOP) of  $M_i$ , on the other hand, lands on point  $p_i$  in the same plane since scaled orthographic projection basically means to project  $M_i$  down to the plane  $K$  at a distance  $Z_0$  from the camera, and to then scale the vector  $P_i$  down to the image plane.

Computing the camera pose assuming a SOP, given the input feature points  $M_i$  and perspective projections of the same points,  $m_i$ , implies an error in the pose since the projections due to SOP would have landed in the points  $p_i$ . To improve the camera pose and get closer to the true one, the algorithm in the next step updates the  $m_i$  points using the pose just found, together with the feature points  $M_i$ . In this way the  $m_i$  points will get closer to where they would have been positioned if the projection was a SOP instead of a perspective projection. Eventually the change

---

<sup>7</sup>The two-dimensional, or 3D if homogeneous coordinates, image points.



in  $m_i$  is small enough to assume that the camera pose has converged. The reason to do this, instead of working with the perspective projection itself, is because it would give us a non-linear, more complicated, problem to solve.

#### 3.3.2 Modern POSIT

In this work, an updated version of the POSIT algorithm described above has been used, called modern posit, [19]. It is based on the same geometrical relations as described for the classic POSIT, but uses homogeneous coordinates (described in next paragraph), allowing the origin of the object’s coordinate system to not be included as one of the feature points, implying that it neither has to be present in the image view. An important note is that the number of non-coplanar feature points has to be at least four to be able to solve the linear system of equations since the matrix  $\mathbf{A}$  needs a rank of 4 [19]. If more than four non-coplanar points are tracked on the object, a POSIT estimation can be more accurate, but if two of them happens to be in the same plane there is no improvement at all, in fact such points can make the algorithm perform worse [14, p. 412]. This is the reason most test environments specified in section 3.2 contain four feature points, to avoid violating the non-coplanar condition. One environment contains ten points to examine possible improvements.

##### 3.3.2.1 Homogeneous coordinates

Homogeneous coordinates, a system of coordinates in perspective geometry, especially useful in for example computer graphics, are described by  $(N+1)$ -dimensional vectors, where  $N$  is the dimension for the corresponding Cartesian coordinates in Euclidean geometry. For example, a 3D Cartesian vector, becomes 4D in homogeneous coordinates and a 2D vector becomes 3D. The “extra” coordinate is typically assigned the letter  $w$ , and basically holds the information needed for how to scale an object in a perspective projection. As an example, one can imagine a 2D projection of a movie on a screen. The coordinates for an object in the image depends on how far away the projector is from the screen. In a 2D example, that distance is the  $w$  value in the 2D object’s homogeneous coordinate representation,  $(x, y, w)$ . If the coordinate vector is divided by  $w$ , the perspective projection of the object is obtained: its 2D Cartesian coordinates on the screen,  $(\frac{x}{w}, \frac{y}{w})$ . Thus, the further away, the greater  $w$ , and the smaller object is presented. For the interested reader, a more thorough description of homogeneous coordinates can be found in Reference [20].

##### 3.3.2.2 Mathematical description

A transformation from 3D-points to 2D-points expressed in homogeneous coordinates is described in Equation (3.1), with  $f$  as the camera’s focal length,  $\mathbf{R}$  (with rows  $\mathbf{R}_1$ ,  $\mathbf{R}_2$  and  $\mathbf{R}_3$ ) and  $T$  as the rotation matrix and translation vector,  $M_0M_i$  as the 3D vector from the origin of the world coordinate system to feature point  $i$ , and the left hand side corresponding to the homogeneous representation of the 2D projection of the feature point  $i$ . This is a perspective projection, represented by a

non-linear system of equations.

$$\begin{bmatrix} w_i x_i \\ w_i y_i \\ w_i \end{bmatrix} = \begin{bmatrix} f\mathbf{R}_1^T & fT_x \\ f\mathbf{R}_2^T & fT_y \\ \mathbf{R}_3^T & T_z \end{bmatrix} \begin{bmatrix} \mathbf{M}_0\mathbf{M}_i \\ 1 \end{bmatrix} \quad (3.1)$$

For a scaled orthographic projection, SOP, the transformation instead looks like in Equation (3.2), where  $s = f/T_z$ .

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} s\mathbf{R}_1^T & sT_x \\ s\mathbf{R}_2^T & sT_y \end{bmatrix} \begin{bmatrix} \mathbf{M}_0\mathbf{M}_i \\ 1 \end{bmatrix} \quad (3.2)$$

Given this, it can be seen that a SOP can be represented as a linear transformation, generally expressed as

$$\mathbf{AI} = \mathbf{x}', \quad \mathbf{AJ} = \mathbf{y}'. \quad (3.3)$$

Here, the rows of the matrix  $\mathbf{A}$  consist of the world coordinate feature points of the object, and  $\mathbf{x}'$  and  $\mathbf{y}'$  are the vectors containing the image coordinates of the projected feature points. The solutions to the linear systems in Equation (3.3) are found by computing  $\mathbf{B}$ , the pseudoinverse matrix of  $\mathbf{A}$ , as in Equation (3.4). An expansion of the linear system in Equation (3.2) is presented in Equation (3.5).

$$\mathbf{I} = \mathbf{Bx}', \quad \mathbf{J} = \mathbf{By}' \quad (3.4)$$

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 \end{bmatrix} \begin{bmatrix} s\mathbf{R}_1^T & s\mathbf{R}_2^T \\ sT_x & sT_y \end{bmatrix} = \begin{bmatrix} w_1x_1 & w_1y_1 \\ w_2x_2 & w_2y_2 \\ \vdots & \vdots \\ w_nx_n & w_ny_n \end{bmatrix} \quad (3.5)$$

In each iteration of the loop, the positions of the projected feature points in the image plane are updated through an update in the scalar values  $w_i$ , given by

$$w_i = \mathbf{R}_3 \cdot \mathbf{M}_0\mathbf{M}_i / T_z + 1 \quad (3.6)$$

where  $\mathbf{R}_3$  is computed from the cross product of  $\mathbf{R}_1$  and  $\mathbf{R}_2$  in the solution to the linear system [18, 19].

### 3.3.3 Advantages and performance

Two main benefits from using POSIT, compared to other algorithms focusing on finding the pose, are the non-expensive computations and that no initial guess is needed. In other algorithms the pseudoinverse matrix has to be computed in every iteration whereas in POSIT the matrix inversion operation is only done once in the beginning. This, in combination with a small number of floating points operations, makes the algorithm way faster and, due to DeMenthon and Davis, “a useful alternative for real-time tracking” [17]. The computational cost neither scales exponentially with the number of feature points, leaving this as an opportunity of

improving accuracy instead. Based on the results from Reference [17], the algorithm typically converges to the true pose already after 4-5 iterations.

There are various algorithms that perform an entire camera calibration to get also the intrinsic parameters of the camera, whereas POSIT assumes that these are already known, and finds the extrinsic parameters, the camera pose, based on these inputs [17]. For algorithms computing the intrinsic camera parameters as well, most seem to not be in compliance with a non-fixed camera position and/or irregular positioning of feature points.

## 3.4 Evaluation of a found pose – error estimation

The estimated camera pose from a camera calibration can be evaluated by measuring the errors in the position and orientation. The position error was in this work computed as the absolute value of the euclidean distance in mm, from the difference between the true position vector and the estimated one. For the orientation error, the matrix norm on the set of rotation matrices was used. The error estimation function computes the least angle of rotation needed to align the coordinate axes of a system rotated by an estimated rotation matrix, with a system rotated by the rotation matrix for the true Euler angles. The computations were based on Equation (3.7) and (3.8), where  $A$  and  $B$  are the rotation matrices defining the two orientations to be compared.

$$W = \log(A^T B) \quad (3.7)$$

$$\text{angular error} = \text{norm}_2(W) \quad (3.8)$$



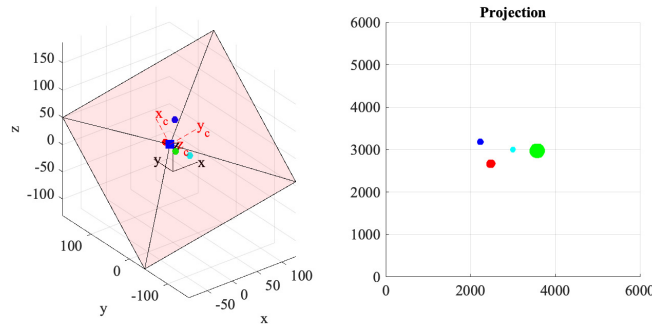
# 4

## Results

In this chapter, the results are presented in terms of testing the forward model for perspective projection from 3D to a 2D image plane, as well as the the inverse model, computing a camera pose from a 2D image and a known 3D world using the POSIT algorithm.

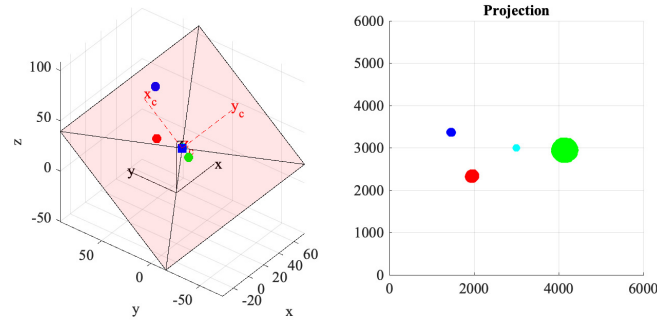
### 4.1 Forward model validation

In Figure 4.1, 4.2 and 4.3 the same environment and camera setup is presented from different angles of view. There are four equally big spheres placed in the 3D space and a camera with its own coordinate system positioned in such a way that all of the spheres are within its field of view, visualized by a transparent pyramid. The rightmost images show the 2D image projection that the camera sees. Comparing Figure 4.1 and 4.2, it can be seen that the whole projection of the spheres is compressed (both scaled down in size of the spheres and in positioning of the spheres in the image plane) by a factor of 2, because of the focal length being 1500 pixels in the first case, and 3000 pixels in the second. The perspective projection is more easily seen in the 3D side view in Figure 4.3, where it can be verified by for example that the closest sphere is the green one, which also appears as the biggest one in the projection, and that the dark blue sphere comes almost as close to the center of the image plane as the red one, even though they are very differently positioned in  $(x_c, y_c)$  coordinates<sup>1</sup>. The distances and sizes in different projections were also examined numerically for verification.

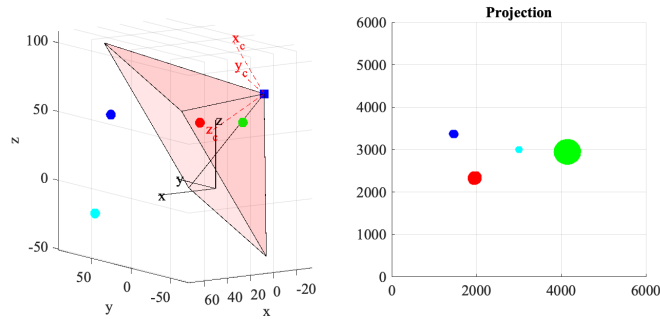


**Figure 4.1:** Perspective projection of four spheres with a focal length of 1500 pixels.

<sup>1</sup>Note that the  $(x_c, y_c)$  coordinates are the sphere's position expressed in the camera's coordinate system, and are not the same as the image plane's pixel coordinates.



**Figure 4.2:** Perspective projection of four spheres with a focal length of 3000 pixels.



**Figure 4.3:** Perspective projection properties shown with spheres of the same size in different positions to the camera.

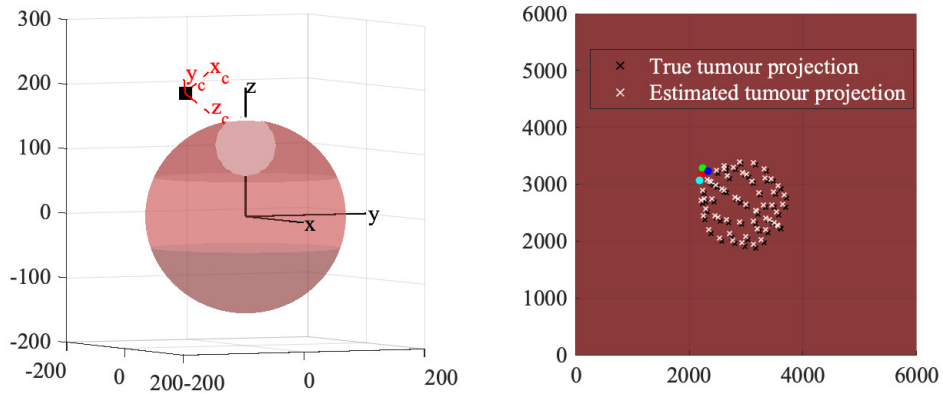
In the following results, the forward model was assumed working, and the markers were only represented by one single point in 3D space each, instead of by spheres. This make their sizes not appear bigger or smaller when they are further away from, or closer to, the camera, as they do in Figure 4.1, 4.2 and 4.3.

## 4.2 Inverse model - evaluation of POSIT

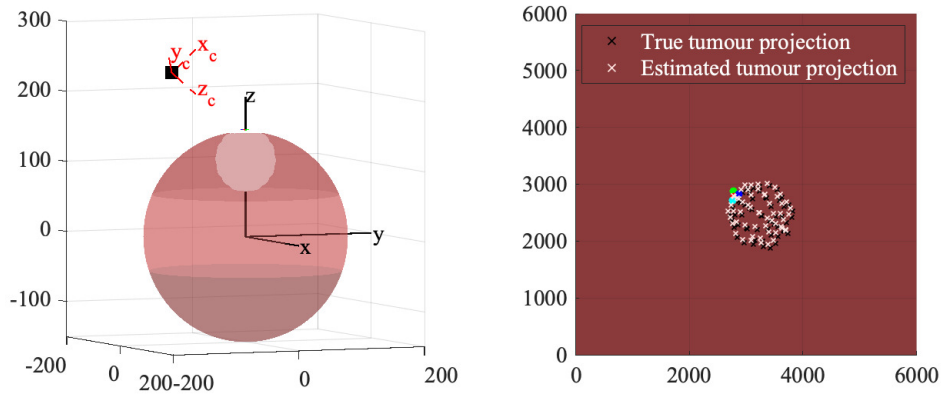
This section presents the result and evaluation from testing the algorithm for finding the camera pose, POSIT. From now on, all results were obtained using a focal length of 3000 pixels, corresponding to approximately 30 mm. There will be a couple of figures similar to the one in Figure 4.4. To the left in those figures, an overview of the 3D environment is shown together with the camera and its coordinate system. The 2D image projections of this 3D scene, obtained from the forward model, are shown to the right in the same figure. The true tumour projection, using the true camera pose in the forward model, is marked by black crosses, and the white crosses correspond to the tumour projected using the POSIT-computed camera pose. The difference between these two projections is caused by the errors from when POSIT estimates the camera pose.

### 4.2.1 Static example

With environment 6 from Table 3.2, and the camera positioned at a distance of 103 and 142 mm respectively, the projections presented in Figure 4.4 and 4.5 were obtained. In both cases the orientation of the camera was defined by the Euler angles  $\alpha_x = 40^\circ$ ,  $\alpha_y = 140^\circ$  and  $\alpha_z = 90^\circ$ . The smaller camera to marker distance of 103 mm, gives an absolute position error of 0.0209 mm, and an angular error of 0.0000694 radians. The greater camera to marker distance of 142 mm, gives an absolute position error of 0.0233 mm, and an angular error of 0.000105 radians. Thus, the closer camera position in this case is more precise.



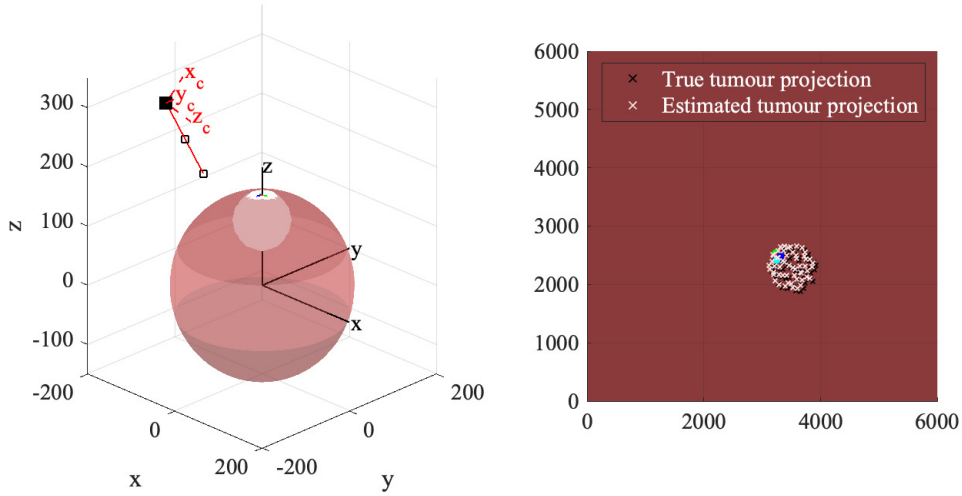
**Figure 4.4:** Environment 6 with the camera positioned 103 mm from the markers.



**Figure 4.5:** Environment 6 with the camera positioned 142 mm from the markers.

### 4.2.2 Translational movement

To evaluate POSIT's performance in different positions, the camera was moved along different paths. In this first test, the path was a straight line, seen in Figure 4.6 in red, moving the camera away from the markers. The camera position is marked by a black square with its coordinate axes ( $x_c, y_c, z_c$ ) in red. The 2D image projections from the forward model are shown to the right in the same figure. The true tumour projection, using the true camera pose in the forward model, is marked by black crosses. The white crosses correspond to the tumour projected using the POSIT-computed camera pose.



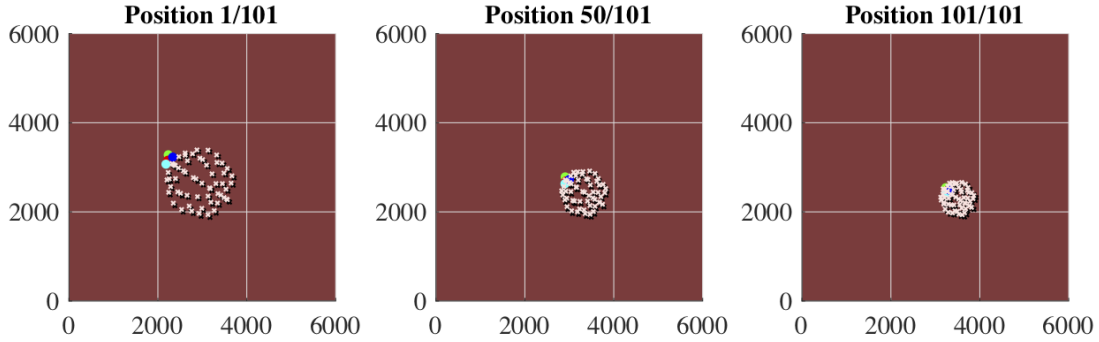
**Figure 4.6:** 3D and projected 2D view of environment 6 and the camera path for translational movement, marked in red in the leftmost image. Three different camera positions: 1, 50 and 101 are marked with black squares along the camera path. The projections to the right are obtained from a fixed camera pose at the end distance of 121 mm, on the linear path, with an orientation defined by  $(\alpha_x, \alpha_y, \alpha_z) = (40, 140, 90)^\circ$ . Because the black and white crosses (tumour projections) do not completely overlap there is an error in the POSIT estimated camera pose.

#### 4.2.2.1 Fixed orientation

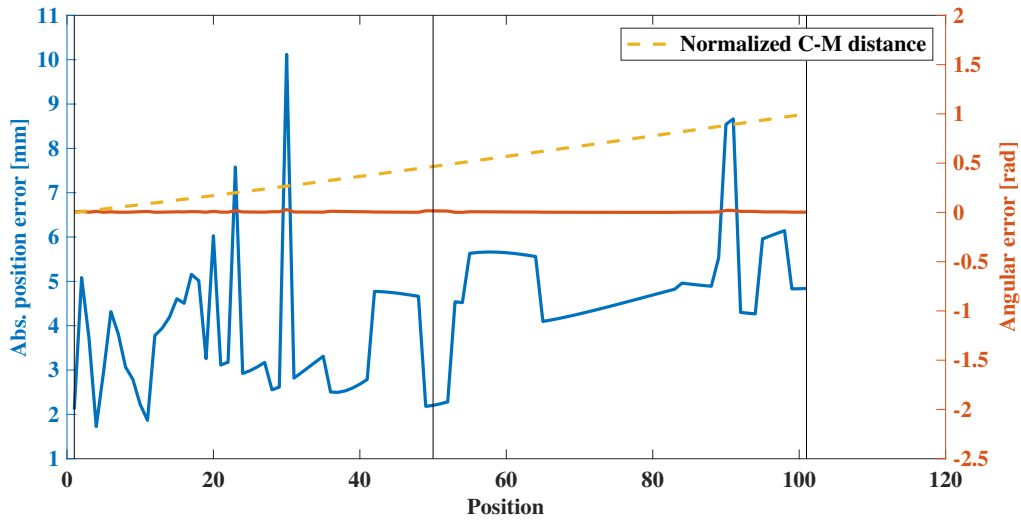
Primary, the camera was moved along the straight line with a fixed orientation of  $(\alpha_x, \alpha_y, \alpha_z) = (40, 140, 90)^\circ$ . The result is shown graphically in Figure 4.7, in terms of the projections at three different camera positions: 1, 50 and 101. These three positions are marked with black squares along the camera path in Figure 4.6. The range of distances from camera to markers starts at 103 mm and ends at 121 mm.

The pose errors along the path are shown in Figure 4.8. The dashed yellow line is the normalized camera-to-markers, C-M, distance. The scale for this as well as the angular error (in radians) is on the right y-axis. The left y-axis holds the absolute position error's scale in mm. The angular error remains constant around 0, meaning that POSIT finds a good approximation to the camera orientation. The

position error fluctuates more, around 3-5 mm of difference from the true position. The general trend seen is although an increase in the absolute position error, when the camera moves further away from the object, just as in section 4.2.1. Such behavior is verified in the paper from DeMenthon in 1995 [17], and is due to the pixel quantization of the camera. If a point lies further away it can move a greater distance before affecting the pixel values, i.e. the image of it, than a point located more close up.



**Figure 4.7:** Environment 6, projections in translational movement with fixed camera orientation. Black crosses represent true tumour position, and white the estimation. The three images are obtained at position 1, 50 and 101 respectively, marked in Figure 4.6 with black squares along the camera path.

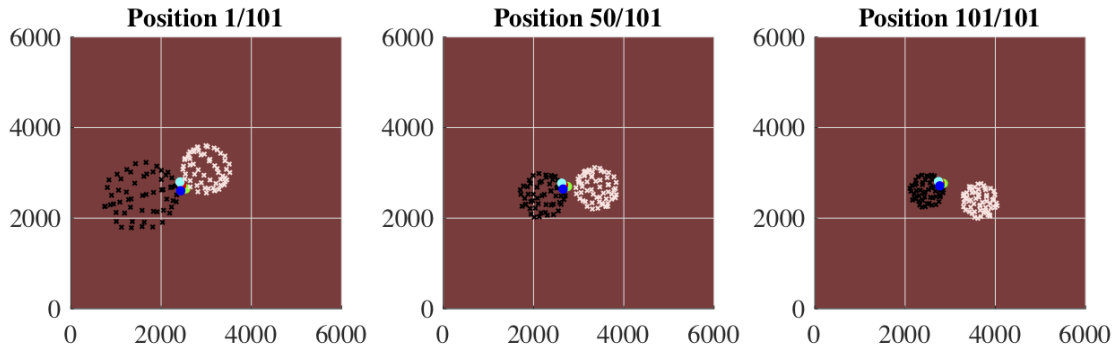


**Figure 4.8:** Errors in estimated camera position (mm) and orientation (radians) for projections of environment 6 in translational movement with fixed camera orientation. Position 1, 50 and 101, corresponding to the three projections in Figure 4.7 and the black squares along the camera path in Figure 4.6, are marked out with black, vertical lines in this plot. The dashed yellow line represents the normalized camera-to-markers (C-M) average distances in each camera position along the path, and shares the right y-axis with the angular error.

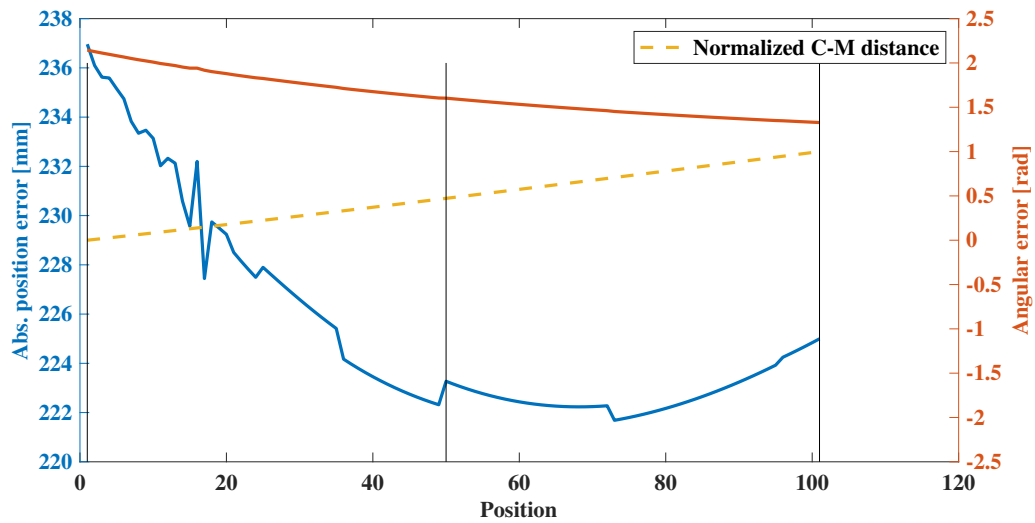
#### 4.2.2.2 Non-fixed orientation

In this test, the camera moved along the same linear path (see Figure 4.6), but its orientation was varied under the constraint that the  $z_c$  axis constantly points at the same (arbitrarily chosen) point just above the liver's surface during the full movement, i.e. the orientation is adapting to each new position. The result is shown graphically in Figure 4.9, in terms of the projections at the three different camera positions: 1, 50 and 101. These three positions are marked with black squares along the camera path in Figure 4.6. The range of distances from camera to markers starts at 115 mm and ends at 121 mm, just as in the previous test.

The pose errors along the path are shown in Figure 4.10. The dashed yellow line is the normalized camera-to-markers, C-M, distance. The scale for this as well as the angular error (in radians) is on the right y-axis. The left y-axis holds the absolute position error's scale in mm. Both the angular error for the camera's orientation, as well as the absolute position error, are in this case decreasing with increased C-M distance. The effect of increasing error with increased camera distance is not seen at all, apart from in the last 20 positions for the absolute position error. This is probably because the system is highly unstable and produces errors in such a large scale. The affect of these errors can be clearly seen in Figure 4.9, where none of the three tumour projections captured are overlapping.



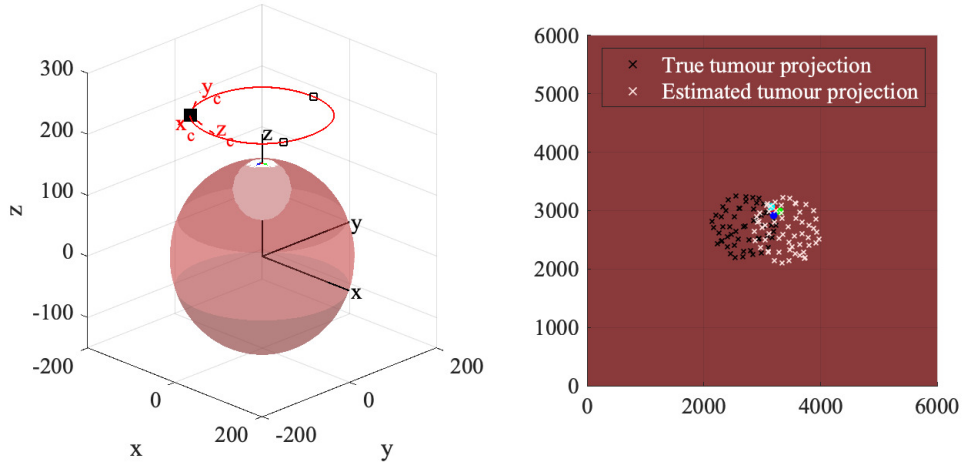
**Figure 4.9:** Environment 6, projections in translational movement with non-fixed camera orientation. The camera instead looks at the same point throughout the whole movement, i.e. the orientation is adapting to each new position. Black crosses represent true tumour position, and white the estimation. The three images are obtained at position 1, 50 and 101 respectively, marked in Figure 4.6 with black squares along the camera path.



**Figure 4.10:** Errors in estimated camera position (mm) and orientation (radians) for projections of environment 6 in translational movement with non-fixed camera orientation. Position 1, 50 and 101, corresponding to the three projections in Figure 4.9 and the black squares along the camera path in Figure 4.6, are marked out with black, vertical lines in this plot. The dashed, yellow line represents the normalized camera-to-markers (C-M) average distances in each camera position along the path, and shares the right y-axis with the angular error.

### 4.2.3 Circular movement

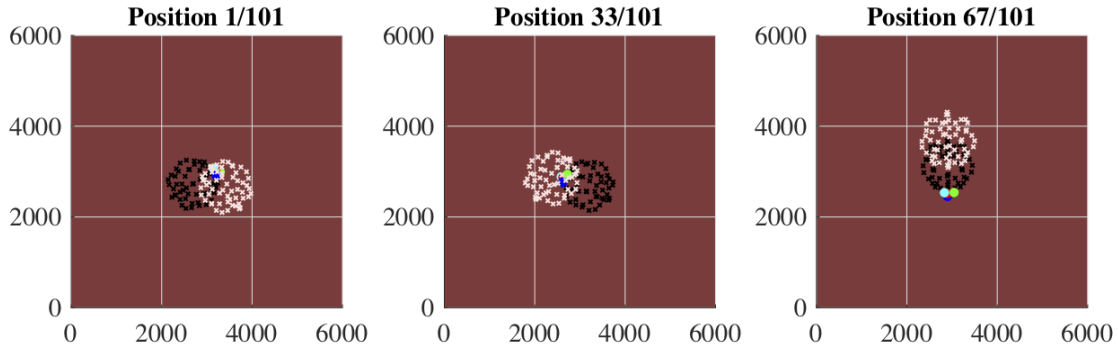
To further examine how the orientation of the camera matters for POSIT's performance, a circular camera path was examined, illustrated in Figure 4.11. Again, the camera's orientation was varied under the constraint that the  $z_c$  axis constantly points at the same point during the full movement, i.e. the orientation is adapting to each new position. The result is shown graphically in Figure 4.12, in terms of the projections at the three different camera positions: 1, 33 and 67. These three positions are marked with black squares along the camera path in Figure 4.11.



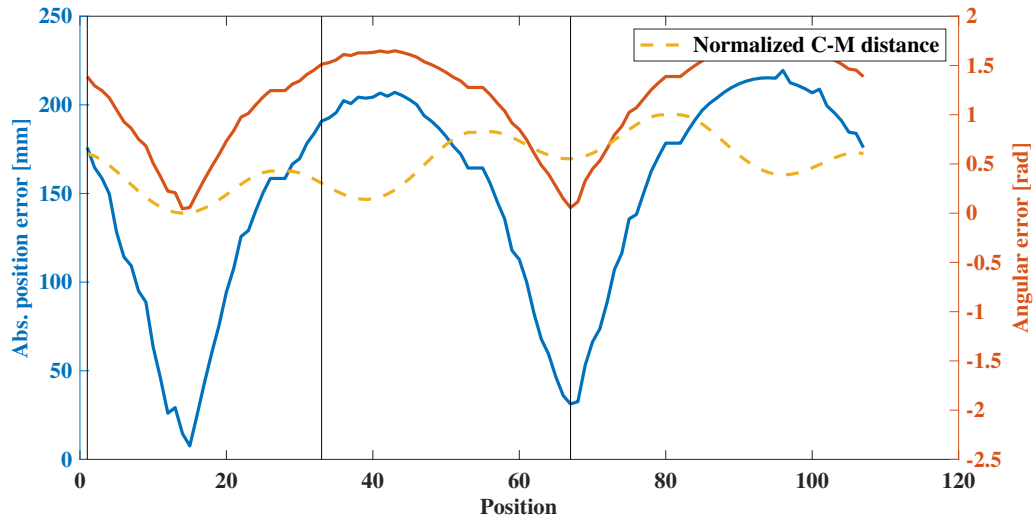
**Figure 4.11:** 3D and projected 2D view of environment 6 and the camera path for circular movement, marked in red in the leftmost image. Three different camera positions: 1, 33 and 67 are marked with black squares along the camera path. The projections to the right are obtained from a fixed camera pose at the end of the circular path, marked as the black square camera position in the leftmost 3D view. Because the black and white crosses (tumour projections) do not completely overlap there is an error in the POSIT estimated camera pose.

The errors from the circular path, seen in Figure 4.13, have some kind of periodical behavior. This might be a cause of the markers' configuration being more or less easy recognized from different angles around the path. It could also be that the camera comes very close to one of the markers at certain positions, deteriorating the camera to markers distance. However, the errors are still in ranges of pretty high values. The position error is almost as high as the error in the previous test along the straight line. The result of these errors are seen in Figure 4.12, where the tumour projection based on POSIT's camera pose overlaps very little with the true projection. It is better than the previous case, but still very poor.





**Figure 4.12:** Environment 6, projections in circular movement with the camera oriented in such a way that the  $z_c$  axis always points to the same point. Black crosses represent true tumour position, and white the estimation. The three projection images are captured at position 1, 33 and 67 respectively, marked in Figure 4.11 with black squares along the camera path.



**Figure 4.13:** Errors in estimated camera position (mm) and orientation (radians) for projections of environment 6 in translational movement with fixed camera orientation. Position 1, 33 and 67, corresponding to the three projections in Figure 4.12 and the black squares along the camera path in Figure 4.11, are marked out with black, vertical lines in this plot. The dashed, yellow line represents the normalized camera-to-markers (C-M) average distances in each camera position along the path, and shares the right y-axis with the angular error.

#### 4.2.4 Environment variation

In this section different environments, i.e. configurations of markers presented as projections in Figure 4.14, from Table 3.2 were examined to see if the performance could improve. The POSIT algorithm has two major limitations due to the assumptions it is based on, listed below. With these two assumptions in mind, the results presented in Table 4.1 and 4.2 can be more easily interpreted, showing how much these limitations affect the usage of the algorithm.

1. The distance between the markers (M-M) has to be much smaller than the distance from the camera to the markers (C-M).
2. The markers have to be non-coplanar.

**Table 4.1:** Accuracy test between the different environments specified in Table 3.2. All environments were tested with a camera to marker distance around 223 mm. For environments A, C and G, POSIT did not converge (A: not to a realistic pose).

Env.	M-M [mm]	Error (Position [mm] / Angular [rad])
A	4.0	158 / 0.459
B	4.8	4.97 / 0.00193
C	5.6	N/A / N/A
D	7.0	4.83 / 0.00264
E	8.14	4.77 / 0.00723
F	9.5	5.63 / 0.00193
G	32.8	N/A / N/A
H	42.8	5.14 / 0.000373
I	45.9	5.16 / 0.000633

**Table 4.2:** Examination of minimal camera to markers distance (C-M) for different environments with different configurations of markers. Abbreviation M-M stands for average distance between the markers. \*For environment A and C, POSIT could not converge to a solution (realistic solution for case A) for any camera distance.

Env.	M-M [mm]	Min C-M [mm]
A	4.0	*
B	4.8	104.18
C	5.6	*
D	7.0	103.13
E	8.14	102.48
F	9.5	104.69
G	32.8	253.5
H	42.8	213.62
I	45.9	98.15

In Table 4.1, various environments were tested on the same case with a camera positioned approximately 223 mm from the markers. The trend is that the greater M-M distance, the smaller angular error for the camera orientation. The absolute position errors seem to remain around 5 mm for all environments. For environment A, POSIT did not converge to a realistic pose. In the cases of environment C and G, POSIT could not converge at all.

In Table 4.2, the minimal camera to markers distance for different environments were tested. None of the environments, except from environment I, could handle C-M distances below 100 mm, where the range of operation during surgery lies, according to Table 3.1.

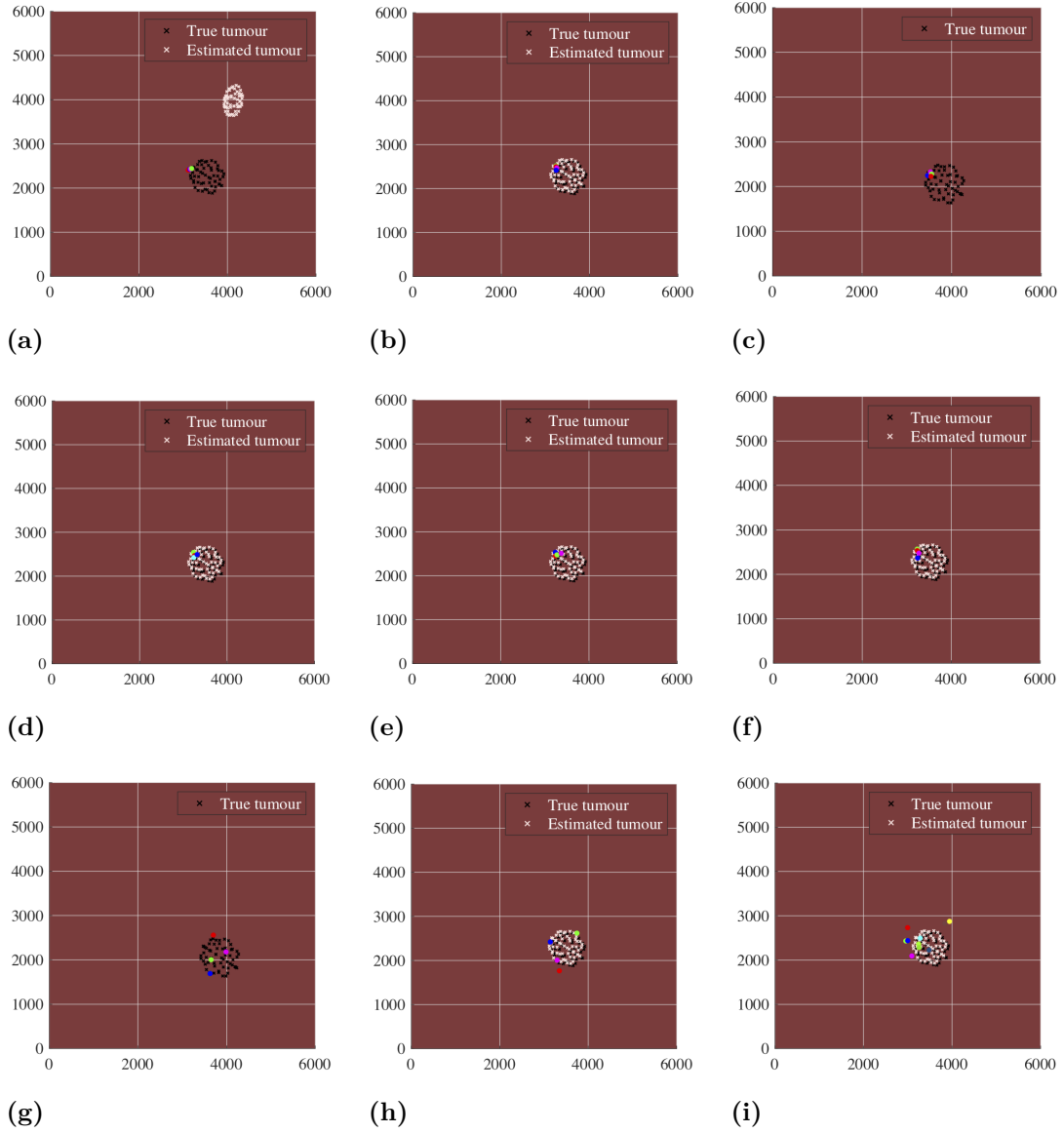
For environment A POSIT could not converge to a realistic solution for any camera distance, seen from the results in both Table 4.1 and 4.2. This is likely due to the markers' configuration, seen in Figure 4.14. For environment C, all four markers had a  $z$ -component with the same number up to the tenth, while other environments differed more. Thus it is assumed that the markers positioned in this environment did not work with POSIT at all, neither in the test presented in Table 4.1, nor in 4.2, because of their coplanar condition.

Results from the environment tests are:

- If markers are placed further away from each other, the camera is limited to not come too close, because of the first POSIT requirement (C-M distance has to be much larger than M-M).
- Markers cannot be positioned too close to each other, because the non-coplanar condition has to be met. The markers also have to be distinguishable in the projection.

#### 4.2.5 Conclusions from POSIT evaluation

The use of POSIT for estimation of camera pose was examined by varying the camera's position, orientation, as well as the configurations of markers by simulating different environments. From these results it can be seen that POSIT does not meet the requirements for this purpose due to the two limitations enumerated above in 4.2.4. Firstly because the algorithm cannot converge when the camera is too close to the markers, at distances that are above the camera's operating range in practice of 3-10 cm. Secondly, the closer the markers are together, the closer the camera can go for convergence of the algorithm. Although, if they are too close together, the algorithm instead experience issues with the second constraint of non-coplanariness. Overall it seems like this last problem is the main reason why POSIT is not a suitable algorithm for solving the inverse problem of finding the camera pose in this application. The liver surface is not curved enough to place the markers on it in a non-coplanar way with the second constraint of that they cannot be too far away from each other.



**Figure 4.14:** Environment configurations for environment A to I. The tumour projections estimated by POSIT (in white) are obtained from a camera pose about 223 mm from the markers, with an orientation defined by the Euler angles  $(\alpha_x, \alpha_y, \alpha_z) = (40, 140, 90)^\circ$ . The tumour projections in black are the true ones, projected using the true camera pose in the forward model.

# 5

## Discussion

An inventory of a laparoscopic liver resection was completed, defining various important factors, such as a limited operation room and camera properties, to take into account when developing an augmented reality tool. The inventory also resulted in specifications for a simulated liver-tumour environment, which can be used for not only this thesis work, but also in future development and algorithm testing.

The forward model used in this work is based on well known camera theory, and according to the results presented, the transformation between coordinate systems as well as the perspective projection seem to work well. In future development of an augmented reality tool, it might be found that the forward model needs to include lens distortions, either because of the existing camera properties, or, if the model shall be adapted to another camera. In the code, there are possibilities to add radial as well as transversal distortions easily, by specifying parameters in a distortion vector which in this work are just all set to zero.

Treating the inverse problem, one algorithm, POSIT, was examined because of its properties: no need for co-planar markers/points, and an algorithm developed for real time use. According to the results, the constraint of non-coplanar points instead turned out to be disadvantageous in this case, because of the liver's radius of curvature not allowing the markers to be enough apart from each other. Due to the second limitation of the algorithm (the distance between markers has to be much smaller than the distance from the camera to them), it was not an option to place markers farther apart to obtain more of a non-coplanar configuration, because then the camera distance grew larger than the operating range of 30-100 mm (from Table 3.1) for the algorithm to converge. In general, the estimated poses were very poor in most situations, even though the camera was further away. The system seemed to be very unstable, changing its performance remarkably with only small perturbations in the simulated camera pose. These limitations, in addition to that the simulated test images did not contain any noise, which in the future might add even more errors to a pose estimation, made the author draw a conclusion of that POSIT is not a suitable algorithm in this problem.

In the paper from 1995 where Daniel DeMenthon presents POSIT for the first time, the algorithm performed very well on distances of interest in this application. Although, it was tested on cubic object with the corner points acting as our markers. Thus, a non-coplanar configuration was always ensured, which, in our belief, was the big problem in the liver-tumour environment, especially because of the liver's

radius of curvature. DeMenthon also mentions that because of POSIT's simplicity, in terms of computations, the algorithm is suitable for real time applications. Our hope and belief is that with today's technology, two and a half decade later, a modification of POSIT avoiding the non-compatible mathematical simplifications done, or in development of a totally different algorithm, computation time might not be an issue.

Even if this thesis only treats the static problem, there will be a future problem in dealing with dynamical changes in tissue structures – both in the pre/intraoperative 3D imaging stage, as well as in a deforming environment during surgery. An advantage with POSIT is that it is not restricted to have a fixed world coordinate system. This means that if the surrounding environment deforms, as long as the markers-tumour relation remains constant, the algorithm will not care about the surroundings. Although, the first requirement has to be that the algorithm works for the markers' positions, even though limited by the liver's geometry. The dynamic problem is briefly discussed further in Appendix.

### 5.1 Future work

With this thesis as a framework, the development of an augmented reality tool in laparoscopic liver resection can continue, and there are yet things to be done. Apart from working on a more suitable and adapted algorithm for solving the inverse problem, there is for example an algorithm detecting the markers in a real laparoscopic image, missing, that has not been examined yet. According to Reference [17] there is a possibility to, in some applications, obtain greater accuracy in camera pose estimation by combining the information from two cameras. In parallel with this, a profitable marker geometry, color and positioning/configuration also need examination. After realizing an algorithm for the inverse problem it shall be decided on design and development of a user friendly AR interface, as informative as possible to satisfy the beneficiaries of the tool. A method for real time error estimation should also be developed, as well as error estimation for when the true camera pose will not be known because it is simulated.

Another part not yet examined is the possibility of not needing a CT in the operation room, but to still obtain a 3D image of the liver with the markers as well as the tumour visible and possible to mark out.

# 6

## Conclusion

The aim of this thesis was to examine the possibility of developing an augmented reality tool for laparoscopic liver resections, through inventory of the surgical procedure and equipment, by modeling a camera in a simulated environment, and by examining an algorithm for camera calibration. In conclusion, a camera model was found and adapted to simulate a laparoscope, based on a pinhole camera model with perspective projection. The model is currently used without distortion, but include the opportunities to add it, if needed.

To calibrate the camera, going from 2D to 3D in the inverse problem, the focal length has to be an input to the system, or calibrated for, since it is a part of the camera matrix needed. A first algorithm to solve the inverse problem, POSIT, was examined. POSIT seem to have some advantages, but applied to this application, the algorithm's limitations hinder use on the length scales and ratios of interest.

A simulated liver-tumour environment was developed based on specifications from the inventory of laparoscopic liver resection. The environment was used to test the POSIT algorithm, but can be used for testing other options as well. It is an adaptive code, with possibilities to add, for example, new marker configurations and camera paths.

In the continuation of this project, another algorithm to solve the inverse problem has to be developed. Furthermore, there are parts not treated at all in this thesis, such as identification of markers in a real laparoscopic video, real time error estimations, and creation of an augmented reality interface to just mention a few.





# References

- [1] *HCC Statistics - Blue Faery*. Accessed on: Oct 7, 2019. American Cancer Society, ASCO Post, CDC, Hepatitis B Foundation, National Cancer Institute and World Health Organization, June 2019. URL: <https://www.bluefaery.org/statistics/>.
- [2] Labadie K., Sullivan K. M., and Park J. O. "Surgical Resection in HCC". In: *IntechOpen* (2018). DOI: 10.5772/intechopen.81345. URL: <https://www.intechopen.com/books/liver-cancer/surgical-resection-in-hcc>.
- [3] Untereiner X. et al. "Short-term and middle-term evaluation of laparoscopic hepatectomies compared with open hepatectomies: A propensity score matching analysis." In: *World Journal of Gastrointestinal Surgery* 8.9 (Sept. 2016), pp. 643–650. DOI: 10.4240/wjgs.v8.i9.643. URL: <https://www.ncbi.nlm.nih.gov/pubmed/27721928>.
- [4] Emspak J. *What is Augmented Reality? - LiveScience*. Accessed on: Sept 16, 2019. June 2018. URL: <https://www.livescience.com/34843-augmented-reality.html>.
- [5] Fang C., Zhang P., and Qi X. "Digital and intelligent liver surgery in the new era: Prospects and dilemmas". In: *EBioMedicine* 41 (Mar. 2019), pp. 693–701. DOI: 10.1016/j.ebiom.2019.02.017. URL: <https://doi.org/10.1016/j.ebiom.2019.02.017>.
- [6] Vávra P. et al. "Recent Development of Augmented Reality in Surgery: A Review". In: *Journal of Healthcare Engineering* 2017 (Aug. 2017), pp. 1–9. DOI: 10.1155/2017/4574172.
- [7] Tang R. et al. "Augmented reality technology for preoperative planning and intraoperative navigation during hepatobiliary surgery: A review of current methods". In: *Hepatobiliary & Pancreatic Diseases International* 17.2 (2018), pp. 101–112. ISSN: 1499-3872. DOI: <https://doi.org/10.1016/j.hbpd.2018.02.002>. URL: <http://www.sciencedirect.com/science/article/pii/S149938721830047X>.
- [8] Lung W. Lau et al. "Laparoscopic Liver Resection with Augmented Reality: A Preclinical Experience". In: *Journal of Laparoendoscopic & Advanced Surgical Techniques* 29.1 (2019). PMID: 30192172, pp. 88–93. DOI: 10.1089/lap.2018.0183. eprint: <https://doi.org/10.1089/lap.2018.0183>. URL: <https://doi.org/10.1089/lap.2018.0183>.
- [9] Hallet J. et al. "Trans-Thoracic Minimally Invasive Liver Resection Guided by Augmented Reality". In: *Journal of the American College of Surgeons* 220.5 (2015), e55–e60. ISSN: 1072-7515. DOI: <https://doi.org/10.1016/>

- j.jamcollsurg.2014.12.053. URL: <http://www.sciencedirect.com/science/article/pii/S1072751515000563>.
- [10] Preim B. and Botha C. “Chapter 18 - Image-Guided Surgery and Augmented Reality”. In: *Visual Computing for Medicine (Second Edition)*. Ed. by Bernhard Preim and Charl Botha. Second Edition. Boston: Morgan Kaufmann, 2014, pp. 625–663. ISBN: 978-0-12-415873-3. DOI: <https://doi.org/10.1016/B978-0-12-415873-3.00018-3>. URL: <http://www.sciencedirect.com/science/article/pii/B9780124158733000183>.
- [11] Aoki T. et al. “Preoperative Tattooing for Precise and Expedient Localization of Landmark in Laparoscopic Liver Resection”. In: *Journal of the American College of Surgeons* 221.5 (2015), e97–e101. ISSN: 1072-7515. DOI: <https://doi.org/10.1016/j.jamcollsurg.2015.07.444>. URL: <http://www.sciencedirect.com/science/article/pii/S1072751515009382>.
- [12] Memeo R. et al. “Innovative surgical approaches for hepatocellular carcinoma”. In: *World Journal of Hepatology* 8.13 (May 2016), pp. 591–596. DOI: 10.4254/wjh.v8.i13.591. URL: <https://www.wjgnet.com/1948-5182/full/v8/i13/591.htm>.
- [13] Cheung T. T. et al. “Pure laparoscopic hepatectomy with augmented reality-assisted indocyanine green fluorescence versus open hepatectomy for hepatocellular carcinoma with liver cirrhosis: A propensity analysis at a single center”. In: *Asian Journal of Endoscopic Surgery* 11.2 (2018), pp. 104–111. DOI: 10.1111/ases.12492. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/ases.12492>.
- [14] Bradski G. and Daebler A. “Learning OpenCV. Computer vision with OpenCV library”. In: (Sept. 2008).
- [15] MathWorks. *What is Camera Calibration?* Accessed on: Sept 23, 2019. URL: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>.
- [16] Taylor Z. *Project 3D into 2D image coordinates using a camera model*. Accessed on: Oct, 2019. Mathworks, Dec. 2014. URL: [https://ww2.mathworks.cn/matlabcentral/fileexchange/48752-project-3d-into-2d-image-coordinates-using-a-camera-model?s\\_tid=FX\\_rc3\\_behav](https://ww2.mathworks.cn/matlabcentral/fileexchange/48752-project-3d-into-2d-image-coordinates-using-a-camera-model?s_tid=FX_rc3_behav).
- [17] Dementhon D. F. and Davis L. S. “Model-based object pose in 25 lines of code”. In: *International Journal of Computer Vision* 15.1 (June 1995), pp. 123–141. ISSN: 1573-1405. DOI: 10.1007/BF01450852. URL: <https://doi.org/10.1007/BF01450852>.
- [18] DeMenthon D. *Code, POSIT (C and Matlab)*. Accessed on: Nov 4, 2019. Aug. 2009. URL: [http://users.umiaccs.umd.edu/~daniel/Site\\_2/Code.html](http://users.umiaccs.umd.edu/~daniel/Site_2/Code.html).
- [19] DeMenthon D. and Davis L. S. “Recognition and Tracking of 3D Objects by 1D Search”. In: *DARPA Image Understanding Workshop* (1993), pp. 653–659.
- [20] Dalling T. *Tom Dalling blog: Explaining Homogeneous Coordinates & Projective Geometry*. Accessed on: Dec 20, 2019. Feb. 2014. URL: <https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/>.

# A

## Appendix

### A.1 Issues in dynamic soft tissue tracking

In liver surgery there are processes, both pre- and intraoperatively (before and during surgery), that will affect the liver's positioning and geometry. It is a general issue that soft tissue, especially if located close to the thorax, is more difficult to keep track of than hard tissue, because of deformations due to, for example, breathing, pressures and touches by tools.

If a preoperative 3D imaging of the liver is performed, that map will not exactly correspond to the 3D environment during the surgery. Pneumoperitoneum, or the inflation of the abdomen with gas (usually CO<sub>2</sub>) affects the liver's position and geometrical shape. If 3D imaging is done intraoperatively, it has to be considered if it was done before or after the pneumoperitoneum. The liver environment is also different if the patient is positioned, for example, on the side compared to on the back. Depending on the time elapsed since imaging was done, the tumour might also have grown bigger. All of these movements and deformations arise an issue when it comes to creating a representative 3D imaging of a liver, even if it is done intraoperatively.

If markers shall be placed on the liver surface, this brings up the question with when they should be placed. In the solution considered in this thesis, the markers have to be present in the 3D image of the liver as well as in the 2D live image from the laparoscope. Markers can be positioned preoperatively, but this implies that the patient has to visit the hospital before the planned liver resection. It is possible to position markers without the need of anesthesia, although a little uncomfortable for the patient. Another risk is that the markers will hide in the time between installation and surgery. Furthermore, when starting to cut in the liver, the 3D view from, for example, a CT-scan is no longer representative since the liver's structure has changed. Although, using an algorithm independent of the placement of the world coordinate system, and only interested in the markers-tumour relation, there might be a possibility to position the markers as close to the tumour that their internal relations as well as to the tumour remains unchanged, or affected very little. It is also known that the liver also holds elastic properties such that if deformed or cut by a tool, the tissue will by itself return to or close to the place where it was before disturbance. How much this helps has to be further examined.