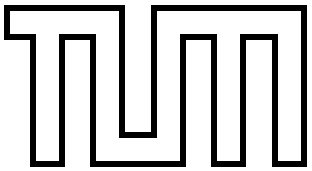


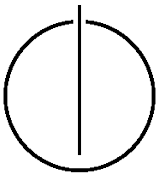
FAKULTÄT FÜR INFORMATIK

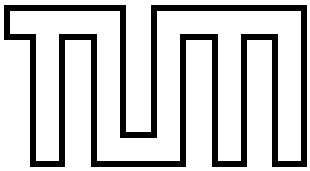
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Diplomarbeit in Informatik

Monocular Real Time Tracking With Passive Circular Markers

Ahmed Matar





FAKULTÄT FÜR INFORMATIK

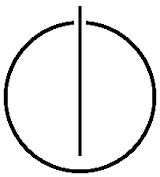
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Diplomarbeit in Informatik

Monocular Real Time Tracking With Passive Circular Markers

Monocular Echtzeit-Tracking mit Passiv-Rund Marker

Author:	Ahmed Matar
Supervisor:	Prof. Dr. Nassir Navab
	Prof. Dr. Michael Friebe
Advisors:	Dipl.Inf. Vasileios Belagiannis
	Dipl.Inf. Benjamin Busam
Date:	August 15, 2014



Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den September 7, 2014 Ahmed Matar

Acknowledgments

I would like to acknowledge the CAMP department for having a friendly helpful environment and providing me with the tools in which i can carry out my thesis.

I would like to thank my advisors, Benjamin Busam and Vasileious Belagiannis, for the guidance and encouragement throughout the thesis. Thank you for making this project such an enjoyable experience. Without you, i could've not reached a deep understanding in the field.

I would also like to thank my supervisors, Prof. Dr. Nassir Navab and Prof. Dr. Michael Friebe for their direction as well as career and future advice, it matters a-lot in this stage of my life.

Last but not the least, a special thank you to my parents, for their support and funding, not only for this year, but throughout my academic career. Your guidance and help is much appreciated and without you, this thesis could never have been started.

Abstract

Image interpretation has attracted a-lot of researchers in the area to make the computer understand what it sees. This problem is relatively easy for the human visual system because we are able to do this very well but under certain circumstances the problem can be even difficult for us. One of the key factors is understanding the geometry behind the image to be able to determine an objects pose and location. A stereoscopy system finds a planes pose by triangulating a point invariant to projective transformations. Throughout this thesis using invariance and projections of geometric shapes, a planes pose can be recovered with a monocular system.

Contents

Outline of the Thesis

Part I: Introduction and Theory

CHAPTER 1: INTRODUCTION

This chapter presents an overview of the thesis and it purpose. Furthermore, it will discuss the history and importance of computer vision.

CHAPTER 2: THEORY

The idea behind the thesis, and some background information.

Part II: Methodology

CHAPTER 3: IMPLEMENTATION

This chapter presents the requirements for the process.

CHAPTER 4: TESTING

The test phases the implementation went through.

CHAPTER 5: CONCLUSION & FUTURE WORK

An overview of the results and future improvements.

Part I.

Introduction and Theory

1. Introduction

As humans we perceive and observe the world around us in three dimensional space easily, but researches were eager to find a way to develop techniques to recover the three dimensional space and appearance of an object from imagery, and from there different mathematical techniques were introduced for corresponding mapping.

The eye can recognize anything under various variation in illumination, viewpoint, expression..etc. According on how robust your algorithm will be to these kinds of conditions in less complexity, the better it is.

One of the basic and critical problems in computer vision is the pose estimation problem which i will provide a possible solution throughout the paper.

To simplify what pose estimation means it is the mapping between the two dimensional points of an image to its corresponding three dimensional coordinates with respect to the world system like an objects rotation and translation.

Pose estimation has many applications including gaming, industrial production, human computer interaction, medical care, security and several others.

Different solutions are introduced till now by the introduction of new mathematical equations or new software or hardware design such as depth sensors [16], new markers [9], or a monocular [17] or a stereo setup [7].

Recently machine learning and artificial intelligence methods like [23, 26, 24] were introduced to proceed in a smart way to compute the unknowns required to map between an image and the world.

1.1. Motivation

There are two ways of setting up your cameras and markers to know their orientation and position *i) Outside-in system* where a set of cameras are at static positions to track the environment in its angle of view or *ii) Inside-out system* where the camera or a set of cameras are moving to track static markers in the environment.

Medical imaging systems such as NDI optical stereo tracker [15, 30] use an outside-in system to track markers attached on hand-held devices (ultrasound, gamma…etc.)(see Fig. (1.1)), the cost of this setup is often beyond the resources of the typical medical community and normally the distance between the stereo-system and the markers is huge making it prone to line of sight interception by the surgeons movement, resulting in lost images and process repetition.

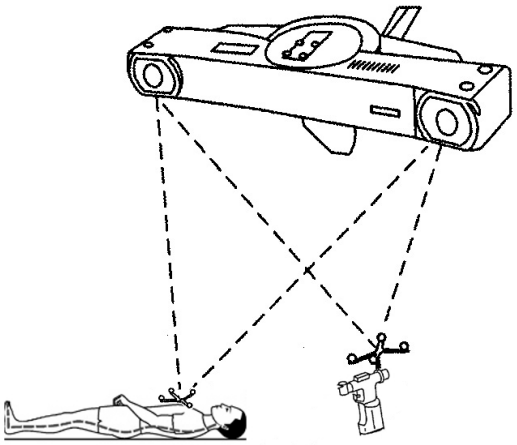


Figure 1.1.: Outside-in stereo tracker

Consequently an inside-out system project is on development, where a camera could be attached directly to the hand-held device being used giving the surgeon more space for movement and lowering its hardware cost, this thesis contributes in experimenting a monocular approach to estimate the camera pose. In the end a decision upon attaching a stereo or a mono system should be reached.

1.2. The Past

At the early time photogrammetry was done by assigning operators to a pair of images to find corresponding features by hand then specify the displacements and feed it to the computer to solve for the position of these points. In 1957, Gilbert Hobrough [18] was the first one to automate the process of stereo correlation for looking for features.

Following this came Larry Roberts [22] in the 1960's his idea was matching in a geometric sense a 3D solid model of an object against its possible projections in the world and so he was able to extract edges and lines, then based on the relative orientations of these lines, he was able to configure out their configuration in three dimensional space.
Since then this set foundation and formulation by which model based vision is still carried out and performing advanced researches in the area.

David Waltz [28] then figured a way from the projection of 3D objects in his case were polyhedral objects could know their three-dimensional configuration from just a 2D image, which is an important step of understanding an objects geometry from just a single view of it.

In the 1980's Rodney Brooks created the ACRONYM system [3] which clarified the role of uncertainty in these geometric analysis, so that small error in camera location or object location could be taken into account in the recognition process.

Complexity started increasing and computers were pretty slow by then, and so the idea of parallel processing and parallelism in the computation of geometric constraints was then identified by Ballard [1] and Feldman [8].

Around 1985 David Lowe [19] and Joseph Mundy [21], made huge impact in hypothesizing the geometric relationships and testing them on a images. Since then basis of illumination, geometry of viewing and projections from single to multiple views and converting them algebraically so they can be computed have been used to develop new unique solutions for pose estimation and model fitting problems.

1.3. Related Work

Problems in computer vision were solved through time some of them are the correspondence, ambiguity, pose-estimation, pattern recognition, background subtraction, efficiency, and accuracy. This section briefly introduces you to recent methods used to solve most of these limitations.

1.3.1. Fiducial Markers

Recently marker tracking has been widely used whether they are 2D or 3D. Types of markers also vary from retro-reflective, simple geometric shapes, natural or fiducial markers. The fiducial markers are special markers encoded with identification code in someway to make the distinguishing process of objects easier also the black and white contrast of the markers make it easier to find in an image, libraries that can be used for marker detection include ARTag [9], ARToolKit [27] and ArUco [11].

1.3.2. Markerless Tracking

A draw back for the previous method is markers have to be there, so researchers developed and still develop another approach which is marker-less tracking also known as photogrammetry [12] or natural marker tracking [14] where given a set of images, identical features can be extracted across the images, then by drawing a ray from the similar features, the intersection is the 3D location of the point, this process is also called triangulation or reconstruction. Across the algorithms each have a different way in selecting the feature points across various illumination or partial occlusions to ensure distinctive photometric properties for reliable pose estimation.

1.3.3. Stereo Tracking

Moreover changes are not only made on markers but on hardware too, the widely used stereo-system [6] has proven its accuracy by finding corresponding features in the left and right images of a scene. Such correspondence match is useful, as for example, in determining the depth feature.
A significant number of publications improved the stereo-system performance [4]. Advancement include having different correspondence methods, methods for occlusion, and real-time implementations. The NDI system [15] is an example of the stereo setup for a medical environment. The setup and priority of an algorithms compromises change across diverse fields to match their cause and needs. .

1.3.4. Mono Tracking

Replacing the stereo-system by a monocular one [2] has been on researchers mind for years. Focus was turned toward methods based on learning of frames [24], back-projecting geometric shapes and using the invariance theory [20], since they contribute enormously in image understanding. Geometric shapes with interesting projections and invariance properties are conics were this thesis and [17] use similar methods for monocular-pose estimation, an alike stereo-system was also developed in [7].

2. Theory

This chapter presents the theoretical background required to reach a deep understanding of our proposed implementation.
The human eye perceives everything in a perspective view which means the objects appearance and dimensions differ relative to the position of the eye, but why do they differ. It is because the eyes are on a plane not parallel to the objects in its field of view. This thesis revolves around achieving parallelism between the planes to make it possible extracting an objects pose with respect to the viewer.

2.1. The Camera

The camera works in the same manner as the eye when you change the focal length, distortion, scaling or angle of view. This is why we have camera calibration techniques to find the quantities internal to the camera that affect the imaging process and explain the plane its on, a good camera calibration is important for camera-pose estimation and reconstructing a world model.
A $3 * 4$ matrix transformation is applied to the camera called projection matrix, this matrix is a composite of camera intrinsic and extrinsic matrices.
Eq (2.1) explains the mathematical operation to project the points from two to three dimensional space where p' is the projection point in the image, A and $[R|T]$ are the intrinisc and extrinsic parameters respectively, and the P' is the 3D point in the world space.

$$p' = A\,[R|T]\,P' \tag{2.1}$$

Going in more details the intrinsic is a $3 * 3$ matrix responsible for linking the pixel coordinates of an image point with the corresponding coordinates in the camera reference frame. Consequently, the f_x and the f_y represent the focal length in pixel units since each camera has a different horizonatal and vertical pixel size, the c_x and c_y are the principal point coordinates (see Eq. (2.2)).

$$A = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

The extrinsic is a $3 * 4$ matrix which defines the translation T_{vector} and rotation R_{vector} of the camera with respect to the world frame (see Eq. (2.3)), for deeper understanding of different camera models you can check it from [13].

$$[R|T] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \tag{2.3}$$

2.2. Marker Perception

Reflective markers are lately being used, these markers simply reflect light resulting in an image that is totally dark with really bright white spots, so it easy to detect the marker in much less complexity compared with detecting colored markers. In the case of colored markers, an example will be demonstrated throughout the thesis.

To explain how color detection occurs, the *RGB* image is converted to *HSV* (Hue, Saturation, Value), the reason behind changing the color space is, HSV separates the image intensity from the color information to be more robust to lighting changes, or removing shadows. By now we can apply a threshold to the image to remove the unwanted colors and then execute a blob detection method to the interesting regions to link them together.

Our theory starts by using the circles geometric properties to determine the pose. The camera perceives the circle as an ellipse similar perception is also between the unit square to a convex quadrilateral due to perspective projection, in most of the cases the two planes are not parallel. If we extend the lines of projection between Π_i *(image plane)* and Π_t *(target plane)* backwards a conic is formed with its vertex at the center of projection (see Fig. (2.1)), a more detailed explanation of how this is formed is if you draw a cone with its base as the circular marker on the target plane, it will intersect the image plane with a conic section of an ellipse with a certain angle.

With the conic section as an ellipse, possible rotations can be calculated to convert the ellipse into a circle. When this conversion is achieved Π_i would be parallel to Π_t. A property was exploited that is unique to a conic that the back-projected section is always a circle, with this you can use some geometry to relate between the back-projected circle and the circular marker on Π_t resulting in finding a possible normal and translation of the circular marker which is the desired pose, this property may also be called an invariant descriptor [20].

An invariant descriptor is a geometric shape that preserves its properties and is unaffected by object pose or transformations.

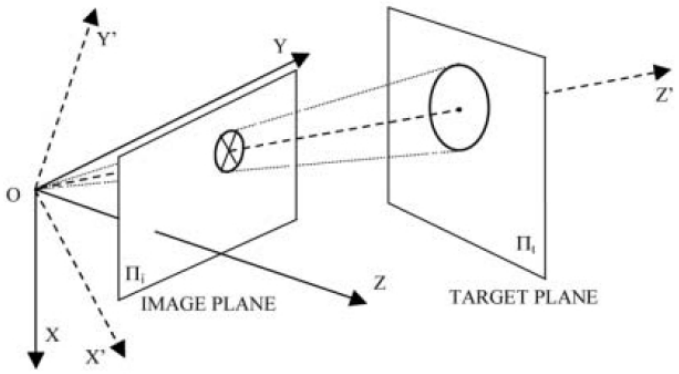


Figure 2.1.: Planes behavior [17]

2.3. Eigen Frame

The eigen values and eigen vectors have an important role in physics and engineering such as building bridges, studying vibrating systems and when estimating the pose of the plane, where in our case you need to keep the properties of the original ellipse matrix. Since when matrices are multiplied by a certain vector they could change direction and become unrecognizable to its original form. And our ellipse goes from one coordinate system to another in order to achieve parallelism.
The preserved direction of transformation is called the eigen vector X (see Eq. (2.4)) and the constant which describes whether the vector is stretched, reversed, or left unchanged is called the eigen value λ.
Geometric interpretations can be expressed in terms of lambda values as you will see in Chapter 3 to retrieve the required angles to transform the ellipse to a circle.

$$AX = \lambda X$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} \tag{2.4}$$

Part II.

Methodology

3. Implementation

3.1. Overview

A brief description about the method would be as a first step fit the points of the marker detected to create the desired conic between the ellipse in the camera plane and the circle on the target plane (see Fig. (2.1)), a rotational transformation to the ellipse should be applied by then to transform it to a circle, which is achieved when the two-planes are parallel, with this property extracting a normal of the target plane is possible which explains its rotational behavior. The rotational transformations could be extracted in terms of eigen-values and vectors, also this is where the ambiguity is formed. Back-projecting the center of the constructed circle to the target plane will form geometric relations between the point and the center of the marker, leading in extracting its normal. By now you extracted the planes normal and its translation which describe an objects pose with respect to the camera.
The method goes through several pipelines as expressed in the figure below and my explanation will go through the same sequence.

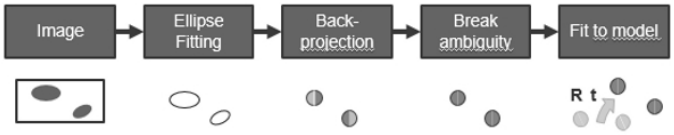


Figure 3.1.: Processing Pipeline

3.2. Setup

This implementation is carried out in matlab since its mathematical notations are concise, simple and powerful. Also its easier in matrix-manipulation, doing specialized functions, and creating plots. Matlab is often slower at execution time, nevertheless it's compensated in the debugging time, and having a fast prototype. Another advantage is its documentation and toolboxes are quick and easy to access, and in the case of our camera calibration we used the matlab camera calibration toolbox.
Camera intrinsic calibration is required A (see Eq. (2.2)), different methods of calibration can be used such as [25, 31].
The marker of choice would be a 2D reflective circular maker as its advantages were explained in Section 2.2. The radius of the markers should be known, also any number of markers can be used as long as at-least 2 or 3 are non co-planar and visible in the camera view, this property is important to break the ambiguity as i will explain in the upcoming pages.

3.3. Ellipse Extraction

After placing the retro-reflective markers and acquiring the image, applying a threshold can easily distinguish the markers as they will have bright white spots as a result.
Fitting of primitive models to image data is a basic task in pattern recognition, and for the data acquired we will need to fit ellipses on the borders of each marker detected. Such method can be done using [10].
After fitting the conic-section in the camera reference frame, the ellipse geometric properties are extracted such as the centers, major and minor radii, and the angle of rotation with respect to the created conic (see Fig. (2.1)).

The geometric properties will need to be converted to an algebraic expression in its normalized form (see Eq. (3.1)) then to its matrix form (see Eq. (3.2)). The ellipse equation is a special case of the conic and based on the x^2 and y^2 coefficients you can extract radii information, the x and y explain how the center of the ellipse is transformed from the origin, and the xy shows the angle of rotation.
Methods from [5] can be used to convert from the geometric to algebraic parameters or the other way around.

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f^2 = 0 \tag{3.1}$$

$$C = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} \tag{3.2}$$

3.4. Pose Estimation

As discussed in Section 2.1 to retrieve the pose you need $[R|T]$, here we will be able to retrieve the normal vectors of Π_t which explains its rotational behavior R around the axes along with the proper translation vector T. A starting point would be multiplying C with A^T and A from the left and right hand-sides respectively this is needed to reveal the ellipses orthnomal basis with respect to the camera frame and adjust the values in C to a common camera reference scale which is a form of normalization.

By observing Fig. (2.1), the orientation of the marker on plane Π_t can be calculated when the image and the target plane are parallel to each other. To achieve this, two successive rotations are applied, the first one R_1 aligns the X and Y axis on Π_i with the axis of Π_t and makes the Z axis go through the center of projection. The second rotation R_2 adjusts the newly found coordinate system $X^{'}Y^{'}Z^{'}$ by rotating it around the $Y^{'}$ axis to become parallel to Π_t, this converts the ellipse into a circle in the $X^{''}Y^{''}Z^{''}$ coordinate system.
While calculating R_2 the two fold ambiguity problem arises (see Fig. (4.2)) resulting in two possible normals $\vec{n}_1$ and $\vec{n}_2$, breaking this ambiguity and choosing the correct normal will be discussed in Section 3.5.

Retrieving the translation vector for the marker will also be ambiguous till we decide on the correct normal.

The vector is calculated by triangular geometry AOO_c (see Fig. (3.2)) so to get the displacement between the center of projection O and the center of the marker O_c in Π_t we need to solve for the distance d and δ (see Fig. (3.2)) an expression in terms of the eigen-values and the marker radius r is calculated (see Alg. (1)).
By now all you have left is to apply the rotational projection on $\begin{pmatrix} \delta \\ 0 \\ d \end{pmatrix}$ and you got the coordinates of the marker center.

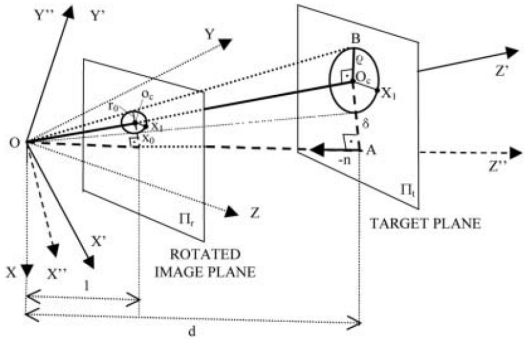


Figure 3.2.: Parallel planes behavior [17]

Data: C matrix in camera frame
Result: normals $\vec{n}_1$ and $\vec{n}_2$, translations $\vec{T}_1$ and $\vec{T}_2$
$\lambda_1, \lambda_2, \lambda_3 =$ eigen-values for C;
$\vec{e}_1, \vec{e}_2, \vec{e}_3 =$ eigen-vectors for C;
if *eigen-values not sorted ascendingly* **then**
  sort $\lambda_1, \lambda_2, \lambda_3$ with respect to $\vec{e}_1, \vec{e}_2, \vec{e}_3$;
end
assuming $\lambda_1 < \lambda_2 < \lambda_3$;
$R_1 = [\vec{e}_1, \vec{e}_2, \vec{e}_3]$;
calculating ambiguous angle to achieve parallelism ;
$\theta_{1,2} = \pm \tan^{-1} \sqrt{\frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_2}}$;
$R_2 = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$;
$R_C = R_1 \cdot R_2$;
$\vec{n}_1$ and $\vec{n}_2 = R_C \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$;
$d = \sqrt{\frac{-{\lambda_2}^2}{\lambda_1 \cdot \lambda_3}} \cdot r$;
$\delta = \sqrt{\frac{-(\lambda_2 - \lambda_1) \cdot (\lambda_3 - \lambda_2)}{\lambda_1 \cdot \lambda_3}}$;
$\vec{T}_1$ and $\vec{T}_2 = R_C \cdot \begin{pmatrix} \delta \\ 0 \\ d \end{pmatrix}$;

Algorithm 1: Retrieving ambiguous pose of Π_t

3.5. Breaking The Ambiguity

Up till now the mathematical concepts are used from [17], however for breaking the ambiguity (see Fig. (4.2)) we follow a different approach. A stereoscopic systems suffers from the ambiguity problem as-well but is solved by comparing features of the image taken from the two cameras in the same frame, a similar approach could be done using the mono-system but you will have to compare over a couple of frames and assuming the camera is always moving. The problem with this approach is it makes you vulnerable that the movement or difference in the pair of frames may not be sufficient to break the ambiguity as well as dropping frames in the middle is not a convenient solution for real time results.

By observing our setup the markers are static in the scene, which allows a constant relation between the markers and each other. To break the ambiguity of a marker M_1 we assume we have model information such as the angle between non co-planar markers, and since you already have the ambiguous normals then calculating the scalar product between a pair will give you 4 possible solutions, following that a comparison with the model information is made then vote for the best fit, and so on with another pair $M_2, M_3 \dots M_n$ with respect to M_1.

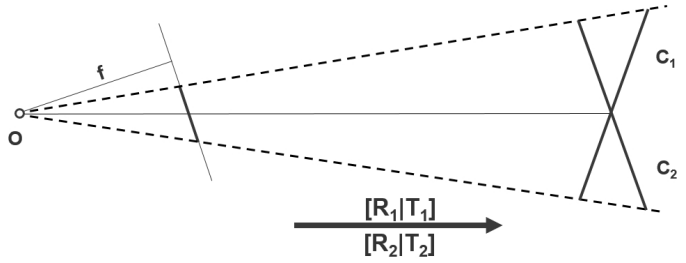


Figure 3.3.: Ambiguity formation per marker

Algorithm 2 explains in a high level language the process to break the ambiguity for a single marker, where M is the specified index of the marker to get its normal, $N_{L_1} and N_{L_2}$ are the lists of the two possible normals for all markers.

Data: M, N_{L_1}, N_{L_2}
Result: $\vec{n}$ correct normal
for $i = 0;\ i < length(N_{L_1});\ i \leftarrow i + 1$ **do**
  if $i \neq M$ **then**
    compute 4 possible scalar products between
    $N_{L_1}(M), N_{L_2}(M), N_{L_1}(i), N_{L_2}(i)$;
    compare with the angle in the model information ;
    score voting from a scale of 4 to 1 with 4 to the closest answer ;
  end
end
return $\vec{n}$ with the highest votes ;

Algorithm 2: Breaking the ambiguity

3.6. Constraints

Calculating the correct ellipse equation is critical since everything is based on it, to convert from the algebraic form (see Eq. (3.1)) to the geometric form is given by [29]

The centers of the ellipse

$$\begin{aligned} x_0 &= \frac{cd - bf}{b^2 - ac} \\ y_0 &= \frac{af - bd}{b^2 - ac} \end{aligned} \tag{3.3}$$

The semi-axes lengths

$$\begin{aligned} a &= \sqrt{\frac{2(af^2 + cd^2 + gb^2 - 2bdf - acg)}{(b^2 - ac)[sqrt(a-c)^2 + 4b^2 - (a+c)]}} \\ b &= \sqrt{\frac{2(af^2 + cd^2 + gb^2 - 2bdf - acg)}{(b^2 - ac)[-sqrt(a-c)^2 + 4b^2 - (a+c)]}} \end{aligned} \tag{3.4}$$

The angle of rotation

$$\theta = \begin{cases} 0 & \text{for } b = 0 \quad \text{and} \quad a < c \\ \frac{1}{2}\pi & \text{for } b = 0 \quad \text{and} \quad a > c \\ \frac{1}{2}\cot^{-1}(\frac{a-c}{2b}) & \text{for } b \neq 0 \quad \text{and} \quad a < c \\ \frac{\pi}{2} + \frac{1}{2}\cot^{-1}(\frac{a-c}{2b}) & \text{for } b \neq 0 \quad \text{and} \quad a > c \end{cases} \tag{3.5}$$

To make sure that R_c corresponds to the correct rotations forming a circle from the ellipse and result in parallelism between planes, it should be in the following form

$$C'' = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & 0 \\ -x_0 & 0 & x_0^2 - r_0^2 \end{bmatrix} \tag{3.6}$$

Where to get to C'' coordinate-frame

$$\begin{aligned} C' &= R_1^T \cdot C \cdot R_1 \\ C'' &= R_2^T \cdot C' \cdot R_2 \end{aligned} \tag{3.7}$$

Correspondences between Equation 3.8 and 3.6 should be established

$$\begin{aligned} x_0 &= \sqrt{\frac{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_2)}{\lambda_2^2}} \\ r_0 &= \sqrt{\frac{-\lambda_3 \cdot \lambda_1}{\lambda_2^2}} \end{aligned} \tag{3.8}$$

4. Testing

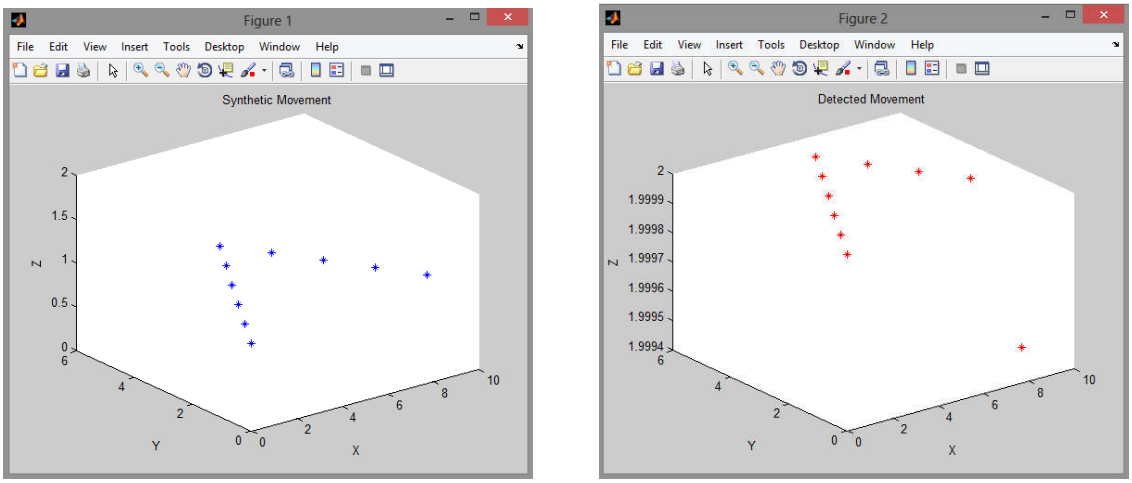
Different experiments are carried out throughout the implementation. This chapter analyzes 3 main approaches used for evaluation starting with synthetic, then intuitive and ending with ground truth comparison. Evaluation is carried out by observing the extracted pose and movement.

4.1. Synthetic

Synthetic tests is data generation computed by mathematical techniques then feeding it to your method and evaluating the results with the generated data.
In our case its a prescribed model with a certain trajectory, and by feeding it to our method you should examine the results by finding a matching movement and model, in this kind of tests we us an identity matrix in the camera intrinsic parameters to avoid any kind of noise.

4.1.1. Circles

In this test we examine matching trajectories. The data is formed by defining a homogenized circle within a unit square of distance 1. The circle center is at the origin $(0, 0, 1)$ and so by applying different transformations across different frames (see Table (4.1)), the detected $\vec{T}$ should change accordingly (see Table (4.2)).



(a) Synthetic Movement of Centers

(b) Detected $\vec{T}$

Figure 4.1.: Comparision between similar trajectories of (a) and (b)

Frame	x	y	z
1	0	0	1
2	1	1	1
3	2	2	1
4	3	3	1
5	4	4	1
6	5	5	1
7	6	4	1
8	7	3	1
9	8	2	1
10	9	1	1

Table 4.1.: Synthetic Translation

Frame	x	y	z
1	0	0	2
2	1	1	2
3	2	2	2
4	3	3	2
5	4	4	2
6	5	5	2
7	6	4	2
8	7	3	2
9	8	2	2
10	8.9975	1.0232	1.9994

Table 4.2.: Extracted $\vec{T}$

4.1.2. Ellipses

The unit square is regarded as a convex quadrilateral due to perspective projection. Furthermore a unit circle can be inscribed in a unit square which projects to be an ellipse inscribed in a quadrilateral. The center of the ellipse can be found by bisecting the four chords joining the points of tangency of this quadrilateral.

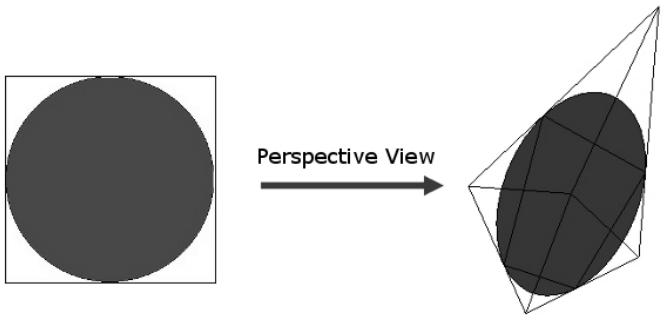
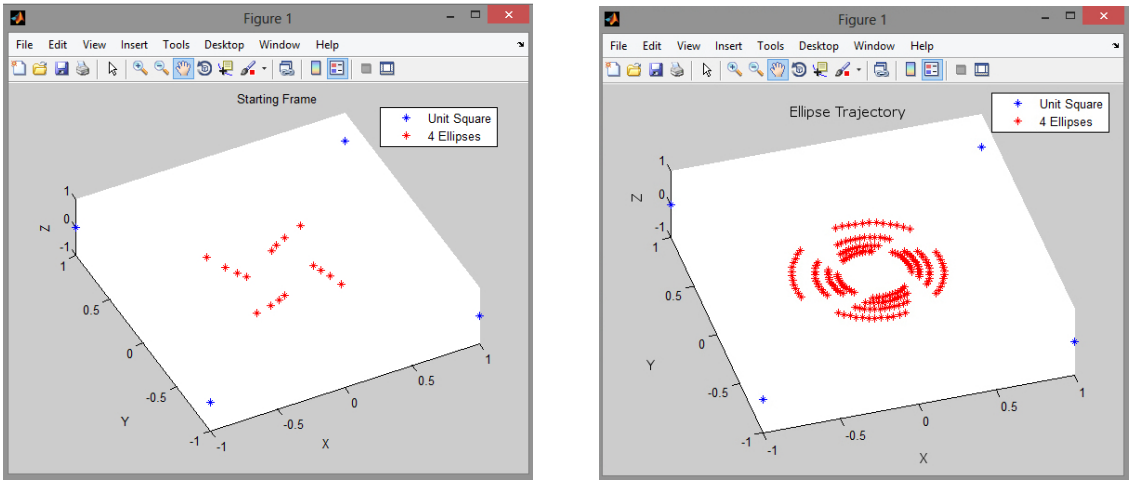


Figure 4.2.: Camera Perspective Projection

In this experiment we use the previous explanation to form 4 circles with the same center but different radii. Following that we apply a unique rotation to each unit circle which backfires on their corresponding ellipses. The circles center is at $(0, 0, 1)$ in homogenized coordinates. From Figure 4.3(a) the outer most 4 points correspond to an ellipse and as you go closer to the center the other 3 are created. Over a couple frames transformations are applied to the ellipses resulting in the trajectory in Figure 4.3(b). The cause of this model is a continuous rotation around the $z - axis$. The purpose of this experiment is verifying the correctness of breaking the ambiguity as well as retrieving the correct center.



(a) Starting position of ellipses

(b) Trajectory applied

Figure 4.3.: Rotations around the $z - axis$ to achieve trajectory

As far as breaking the ambiguity is concerned, model information is acquired through a static rotation around the $x - axis$ expressed in the θ column in Table 4.3.
Moreover we translate the centers by the amount in Table 4.3 and look for matching results.

Circle	x	y	z	θ
1	0	0	2	-10
2	0	0	3	0
3	0	0	4	10
4	0	0	5	20

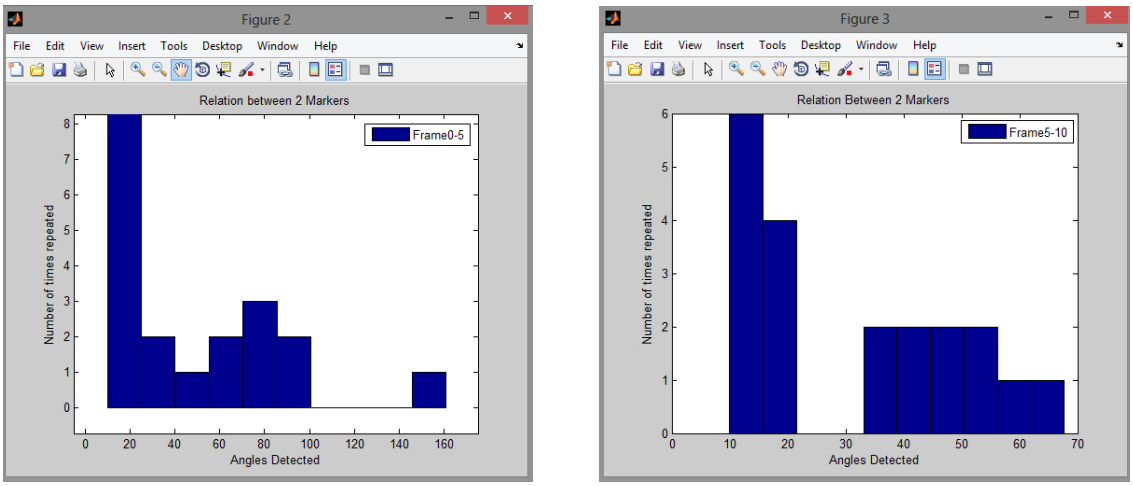
Table 4.3.: Center Transformation

Circle	x	y	z
1	-0.0018	0.1031	2.9982
2	0	0	4
3	0	0	5
4	0	0	6

Table 4.4.: Extracted $\vec{T}$

While breaking the ambiguity (see Sec. (3.5)) 4 possible solutions arise between each pair of markers, consequently by looking at the θ column in table 4.3, retrieving the relation between the pair is viable by finding the difference between the angles and comparing it with the extracted solutions.

For the sake of this test we observe the behavior of the resulting angles between the pair 1 and 2, which has a relation of $10°$. Data is collected over an interval of 5 frames, and the following histograms are formed



(a) Frames 0-5 (b) Frames 5-10

Figure 4.4.: Recording possible angles across frames

From Figure 4.4(a), you can notice that a certain combination has a consistent result for the correct angle which is $10°$, and based on that you can break the ambiguity without referring to model information.

Nevertheless movement between certain frames are sometimes insufficient to break the ambiguity. By observing the results in Figure 4.4(b), two possible peaks are spotted, this is where previous model information comes in handy to detect the true normal.
In our implementation we break the ambiguity each single frame by keeping a score of the voted normal between a marker and its permutations with the others.

4.2. Intuitive

Synthetic data were highly successful, which encouraged to have a working demo where you could intuitively decide if the data collected is right or wrong. This demo is a RGB blob detector (see Sec. (2.2)), where you can take the borders of the blob and fit them to be an ellipse, then continue the computations to extract the $\vec{T}$.
On the other hand $\vec{n}$ cannot be decided since we do not know how the markers are related to each other, so we used a non co-planar setup for simplicity.
In this experiment we used a calibrated webcam to detect 3 markers red, green, and blue, then plot their results accordingly.
Accuracy and precision cannot be measured since having the ground truth of points is absent.

Deciding intuitively is based on moving the markers on a specific pattern and checking if the graph changes accordingly. In our case we tried a square pattern (see Fig. (4.5)), the graph describes the detected $\vec{T}$ of the centers of the marker.

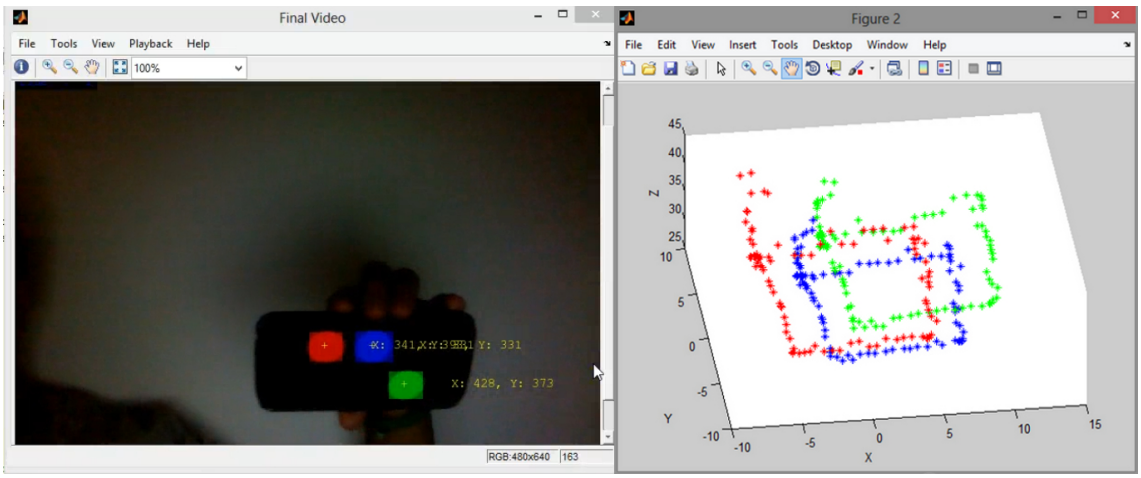


Figure 4.5.: Square pattern by RGB detector

Another possible evaluation can be calculating the deviation of the distance between a pair of markers, since the markers move together with the the same transformation then the distance between them is rigid. In a best case scenario the deviation should be a $zero$. The measured deviation in Table 4.5 is between the red and the $green$ markers over $150 frames$.

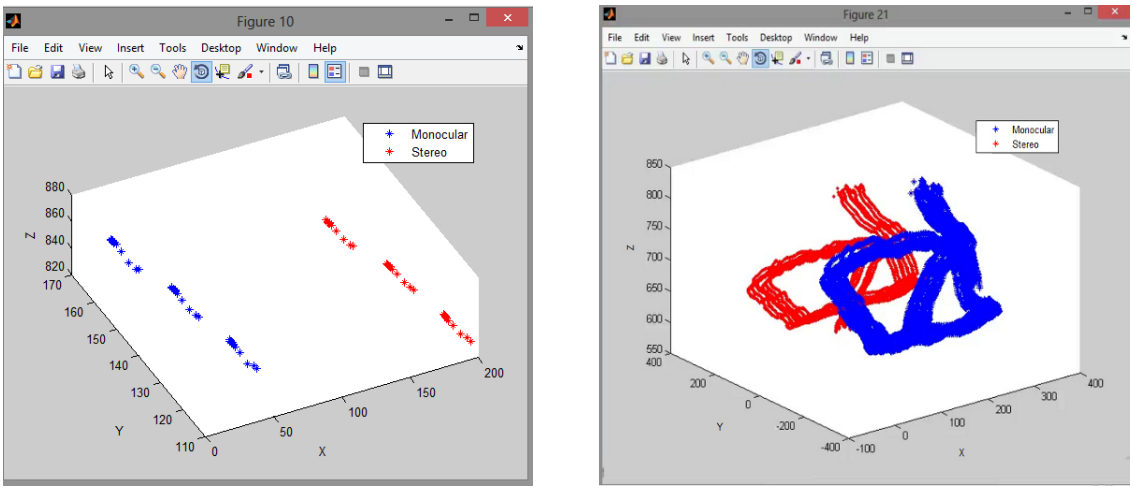
x	y	z
0.2502	0.1538	1.1736

Table 4.5.: Standard Deviation of RGB Detector

4.3. Stereo Comparison

In the third stage of testing, ground truth data is gathered from a stereo-system which is sub-millimeter accurate. The red trajectory in Figure 4.6 represent the marker-motion detected by the stereo-system. The points plotted are the detected $\vec{T}$ of the marker centers.

Our system takes as input from the stereo-system the camera intrinsics as well as properties of the detected ellipses such as the centers in pixel coordinates, major and minor radii, and the angle of rotation, then preforms its computation and plots the extracted $\vec{T}$ as the blue trajectory.



(a) 3 Marker Dataset

(b) 8 Marker Dataset

Figure 4.6.: Monocular result compared with stereo-system

Our theory and logic in extracting the pose is for sure correct, nevertheless this test exploited a missing translation in our implementation since the two graphs did not completely overlap, however same motion and trajectory is detected.
A possible explanation for this shift is the axes are not accurately aligned with the target plane, resulting in a shift in all values. Another theory would be with the introduction of camera intrinsics, the distortion factor should be taken into consideration when calculating $\vec{T}$.

Standard deviation can still be calculated between a pair of markers throughout continuous movement as the distance between them is always fixed.

x	y	z
0.1502	0.1438	0.6396

Table 4.6.: Standard Deviation between a pair of markers

5. Conclusion & Future Work

The purpose of this thesis was experimenting a monocular approach on static non coplanar circular markers to estimate their pose. The math techniques to do this were used before but we proposed a new way to break the ambiguity by voting for the proper normals with respect to the given model. This model is based on the relations between the non-coplanar markers.

Synthetic tests conducted that the method extracts correct normals after going through the voting system, as well as correct translation.

Trajectories are extracted correctly while comparing our approach with the stereo-system nevertheless drawbacks arose with the introduction of camera intrinsics, where a non justified translation is missing for the graphs to overlap (see Fig. (4.6)).

Future improvements include model learning from frames, where instead of knowing ahead how the markers are aligned, an algorithm could observe the first couple of frames to decide on a possible relation between them (see Fig. (4.4(a))). However this requires the camera to be in continuous movement or the voting would be in close proximity to each other resulting in choosing the wrong relation.

Moreover identifying the markers in the RGB test was easy, but for the stereo-system we take as input a text file with the detected markers over a couple of frames. We assign the first frame as reference for marker ordering. The assumption is marker information will come in the same order in all upcoming frames ignoring possible permutation of the data.

A solution to that is to provide a prediction margin, if the marker lays in the next frame within this margin then its the same marker, if not possible exclusion from the voting system could me made since permutation could have happened which might mess up your defined model. This a promising solution for a continuous smooth movement, yet defects occur if exposed to sudden movement.

Finally, conversion of the code from $matlab$ to C++ would be convenient since computer vision libraries are widely available in that language, which may allow researchers develop the method more or add innovative ideas to it.

Bibliography

[1] Dana H. Ballard, Geoffrey E. Hinton, Terrence J. Sejnowski, et al. Parallel visual computation. *Nature*, 306(5938):21–26, 1983.

[2] H. Bassmann and Ph W. Besslich. Monocular computer vision. In *Image Processing and its Applications, 1989., Third International Conference on*, pages 107–111, 1989.

[3] Rodney A. Brooks, Russell Creiner, and Thomas O. Binford. The acronym model-based vision system. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'79, pages 105–113. Morgan Kaufmann Publishers Inc., 1979.

[4] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, Aug 2003.

[5] N. Chernov. Conic fitting algorithms, 1986. Retrieved August, 2014, from `http://people.cas.uab.edu/~mosya/cl/MATLABconics.html`.

[6] I.J. Cox. Stereoscopic computer vision system, 1995. US Patent 5,383,013.

[7] Song De Ma. Conics-based stereo, motion estimation, and pose determination. *International Journal of Computer Vision*, 10(1):7–25, 1993.

[8] Jerome A. Feldman and Jerome A. Connectionist models and parallelism in high level vision. In *Human and Machine Vision*, pages 178–200. Academic Press, 1985.

[9] M. Fiala. Designing highly reliable fiducial markers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(7):1317–1324, 2010.

[10] M. Fitzgibbon, A. W.and Pilu and R. B. Fisher. Direct least-squares fitting of ellipses. 21(5):476–480, May 1999.

[11] S. Garrido-Jurado, R. Munoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marin-Jimenez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47:2280 – 2292, 2014.

[12] Mayer H. Object extraction in photogrammetric computer vision. {*ISPRS*} *Journal of Photogrammetry and Remote Sensing*, pages 213 – 222, 2008.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[14] S. Hinterstoisser, Selim Benhimane, and N. Navab. N3m: Natural 3d markers for real-time object detection and pose estimation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7, 2007.

[15] Northern Digital Inc. Ndi polaris system, 1991. `http://www.ndigital.com/medical/products/polaris-family`.

[16] Kourosh Khoshelham. Accuracy analysis of kinect depth data. In *ISPRS workshop laser scanning*, volume 38, page W12, 2011.

[17] Diego López de Ipiña, Paulo R. S. Mendonça, and Andy Hopper. Trip: A low-cost vision-based location system for ubiquitous computing. *Personal Ubiquitous Comput.*, pages 206–219, 2002.

[18] H.G. Louis. Automatic stereoplotting system and method, August 18 1964. US Patent 3,145,303.

[19] D. G. Lowe. Perceptual organization and visual recognition. Kluwer Academic Publishers, 1985.

[20] J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. Artificial intelligence. MIT Press, 1992.

[21] Joseph L. Mundy and Andrew Zisserman. Geometric invariance in computer vision. chapter Projective Geometry for Machine Vision, pages 463–519. MIT Press, 1992.

[22] L.G. Roberts. *Machine Perception of Three-dimensional Solids*. Its Technical report. MIT Document Services, 1963.

[23] E. Rosten and T. Drummond. *Machine Learning for High-Speed Corner Detection*, pages 430–443. Artificial intelligence. Springer Berlin Heidelberg, 2006.

[24] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, pages 1161–1168, 2005.

[25] Roger Y. Tsai. Radiometry. chapter A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, pages 221–244. Jones and Bartlett Publishers, Inc., 1992.

[26] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition. CVPR*, volume 1, pages I–511–I–518 vol.1, 2001.

[27] D. Wagner and D. Schmalstieg. *Artoolkitplus for pose tracking on mobile devices*. 2007.

[28] David Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*. McGraw-Hill, 1975.

[29] Eric W. Weisstein. "ellipse." from mathworld–a wolfram web resource., 1996. `http://mathworld.wolfram.com/Ellipse.html`.

[30] Andrew D. Wiles, David G. Thompson, and Donald D. Frantz. Accuracy assessment and interpretation for optical tracking systems. pages 421–432, 2004.

[31] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1330–1334, 2000.

List of Figures

List of Tables