Design of an Invariant-based World Model for an Autonomous Football Table

Jebbink, K.S.

Award date:
2022

Mechanical Engineering
Control Systems Technology

Systems & Control

Design of an Invariant-based World Model for an Autonomous Football Table

Kevin Jebbink
0817997

Thesis Supervisor:	M.J.G. van de Molengraft
other Supervisor:	H.P.J. Bruyninckx
Advisor:	J.P.F. Senden

April 22, 2022

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date 4/22/22

Name K.S. Jebbink

ID-number 0817997

Signature 

Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Design of an Invariant-based World Model for an Autonomous Football Table

Kevin Jebbink, Jordy Senden, and René van de Molengraft
Department of Mechanical Engineering
Eindhoven University of Technology
Eindhoven, The Netherlands

***Abstract*—Robots using a world model of its surrounding environment have this world model typically exactly parameterized. This can be very useful mathematically speaking, but it can also be a limiting factor, as these world models are not robust against variations in the parameters. This paper describes an approach of a design of a world model which is robust against variations in the environment. By designing the world model such that the robot skills are based only on a small set of invariants, variations can be embedded in the world model. This means that the proposed world model will perform correctly with any robot that operates in an environment in which these invariants hold, even if certain parameters, such as exact geometric distances between objects, differ. The proposed model is demonstrated on an autonomous football table. This paper describes how a described set of invariants for a football table can be used to automatically calibrate a camera hanging above the field for object tracking, as well as calibrate the world model. Experiments will demonstrate that the performance of the autonomous football table is within the desired limits, and that the new world model is robust against variations, such as changing camera position and football tables with different dimensions.**

I. Introduction

A robot is an autonomous machine that is able to observe its environment through the use of sensors, and then use these observations to influence its environment. The local environment that a robot is able to influence is hereafter referred to as the *workspace* of the robot. For a robot to be able to operate in its workspace, spatial knowledge about this workspace is required. This is done by creating a spatial model of the workspace which the robot can use. This spatial model as well as the perception and control affordances of the robot are hereafter referred to as the *world model*. The spatial model is typically exactly defined in Euclidean space, which means that every object in the workspace has its pose expressed in standard units of measurement in respect to a reference point. This is useful, as it creates a clear mathematical description of the workspace from which is possible to derive dynamical models and which can be used to optimize the task a robot needs to perform. Take, for example, a robotic arm at an assembly line. The spatial model of the workspace of this robot is very specifically defined within the robot. The task this robot needs to perform is very predictable and repetitive, and can be easily optimized.

However, such a world model also has downsides. The data from the sensors need to be converted to a metric value in Euclidean space. If the sensor is not calibrated perfectly, this conversion will be off. Furthermore, the created world model is only functional for one specific situation, and changes in the workspace or sensors means that the world model needs to be updated to account for this. Such world models are not robust against variations. Using an Euclidean world model, while mathematically very useful, can thus also be limiting. Consider for example a tomato-picking robot. It is impossible to describe the geometric positions of the tomatoes on a tomato plant in a world model for such a robot, as each tomato plant will have variations. However, all tomato plants do have a similar layout. As such, if a more abstract world model is used, one which is based on the invariant topological relations of the tomato plant, then a robot using such a world model would be able to pick the tomatoes regardless of the exact geometric positions of the tomatoes on the tomato plant. A robot using such a world model would in fact be more similar to a human. A human does not use exact geometric parameters to calculate its actions, but instead focuses on relations between objects and relative distances. This paper proposes a world model which does not use a geometric map with exact metric distances of objects in the workspace, but instead uses a priori and invariant knowledge of the layout of the workspace (which will be referred to as the *topology* hereafter) in which the positions of objects are described relative to the workspace in dimensionless ratios. Such a world model could be described as an invariant-based world model.

The proposed world model will be implemented on the autonomous football table EUTAFT[1]. An autonomous football table is a football table where one side of the football table contains automated rods which are controlled by a built-in computer. Several similar projects have been realized with different solutions for ball tracking, such as [1], [2], and [3]. On EUTAFT, a high speed camera is connected to the computer, which is used to localize and track the ball. Based on the estimated location of the ball in the image of the camera, the computer moves the automated rods such that one of the puppets on the rods is able to intercept and kick the ball. In previous work on EUTAFT, such a system was developed in [4]. In this approach, the manually calibrated camera was located perpendicular to and directly above the table (see figure

[1] http://cstwiki.wtb.tue.nl/index.php?title=Autonomous_Football_Table

1). The pixel location of the ball is converted to a metric position on the field, which is then used to determine a correct rod response. The ball was detected using its red color. Rod response is controlled using encoder sensors in the actuators.

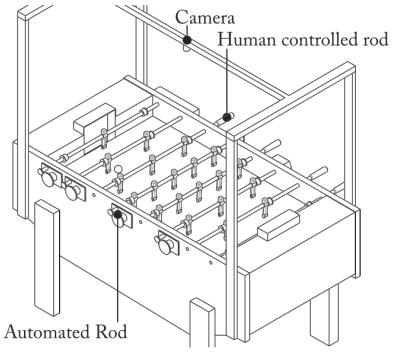


Fig. 1: Schematic representation of the complete set-up (figure from [5])

While EUTAFT performs well using this approach, there are several drawbacks. Several manual calibrations are necessary, such as a camera calibration using a checkerboard (checkerboard calibration described in [6]), a color calibration where the software is 'taught' which colors correspond to red, white, black, and yellow, and a calibration were the user manually selects the pixel locations of the puppets as well as the center-line. Such calibrations can be tedious for the user. Furthermore, several parameters are 'hard-coded' in the world model and are based on the exact dimensions of the EUTAFT set-up. Such parameters are for example the pixel-to-meter ratio which is used to convert pixel measurements to meters, the metric length of the strokes of the rods, the metric length of the reach of the puppets, as well as the specific color value of the ball. Using such 'hard-coded' parameters is limiting, since variations in the set-up require manual adjustments to the world model. Lastly, this world model is not equipped to handle changes in camera position. Even small changes, which can happen during active play, can already result in reduced performance. At these instances EUTAFT needs to be re-calibrated. The camera is also limited to a perpendicular position right above the table at a specific fixed height.

The proposed new approach will automatically estimate the parameters of the world model without the need for manual camera calibrations and manually selecting puppet locations and markings. This world model will be continuously updated. It is hypothesized that the proposed approach is robust against variations, which means that in this approach the camera does not need to be fixed to the exact position above the football table and may move during active play, and that this approach could be universally implemented on any football table, irrespective of table dimensions and colors.

Much research has been done on automated camera calibration, such as in [7] and [8]. These are used as a basis in section IV. [5], [9], [10], [11], and [12] were used to study different image segmentation techniques, which were used to create the identification of the puppets and ball in the image as designed in III. An important goal of software is to track the ball. Previous work on EUTAFT, as well as several other automated football tables, achieve this using a camera directly above and perpendicular to the table as in [2], [3], and [4], though all use manual camera calibration and a convenient and fixed position to place the camera. The non-fixed camera means that estimation of the 3d scene is necessary. [13] and [14] were used to achieve this in section IV.

Section II describes the software architecture of EUTAFT using the proposed new world model. This section will describe the foundation of the new world model, the software *functionalities* that result from this world model, and the way these functionalities interact with each other. Section III and IV describe how these functionalities are implemented, with section III focusing on the object detection of the ball and the automated puppets (hereafter referred to as *foosmen*) in the camera images, and with section IV focusing on the modelling of sensors and the world model and the estimation of the parameters of this new world model. In section V experiments are conducted and its results are presented. In section VI the results are discussed, and in section VII conclusions are made and recommendations for future work are made.

II. Software Architecture

The proposed world model is based on fixed and pre-known relations between the objects on the football table. While football tables come in all kinds of different configurations, in this work the following assumptions are made, which hold true for most general football tables:

- The football table consists of 22 foosmen, 11 per team.
- These 22 foosmen are fixed to 8 parallel rods, which are all spaced apart by the same distance (distance d_r).
- The 22 foosmen can only translate in the lateral direction and rotate around its own axes.
- The 22 foosmen are mounted on the eight rods, per rod from left to right, as follows: one keeper team 1, two defenders team 1, three attackers team 2, five midfielders team 1, five midfielders team 2, three attackers team 1, two defenders team 2, one keeper team 2.
- The foosmen on one rod are all spaced apart by the same fixed distance (distances δ_d, δ_m, and δ_a), and there is no angular offset between the foosmen on one rod.
- The set of foosmen on all rods, except the keeper rods, can be moved all the way from one end of the field to the other end (resulting in stroke lengths s_d, s_m, and s_a).
- The stroke lengths s_d, s_m, and s_a are longer than their corresponding foosmen spacings δ_d, δ_m, and δ_a, otherwise the table would have 'deadspots' (deadspot being defined as an unreachable ball position).
- The goal of one team is located behind the keeper of the other team. The size of the goal is equal to stroke of the keeper foosman (stroke length s_k). The length of this stroke is limited via equally distanced 'stoppers' on the keeper rod (this distance is δ_k).
- The length of the foosmen is such that the foosmen hover slightly above the field when they are orientated straight

up, and that the feet of opposing foosmen can almost touch when orientated at 90°(so as to decrease the amount of 'dead spots' to a minimum).
- Each team has an unique specific team color, which differs from the surrounding colors of the field, in order to see which foosman belongs to which team. The foosmen are uniformly colored according to their team colors.
- The diameter of the ball is in a similar range as the width of the foosmen, and the ball has an uniform color which is different from the two team colors as well as the field color.

These assumptions describe the topology of the foosmen, which forms the basis of the new world model, and are hereafter referred to as *topology invariants*. The decisions the software need to make are based on the location of the ball within the topology of the foosmen. The topology can be described as a rectangular plane with a length of d_l and a width of d_w. The coordinate system used in the world model is based on this plane. When the axis of the coordinate system are aligned with the plane (the length d_l corresponding the x-axis and the width d_w corresponding to the y-axis), then the positions of the foosmen as well as the ball can be described in fractions of these two constants. In this coordinate system, the four corners of the topology of foosmen will then be $(0, 0)$, $(d_l, 0)$, $(0, d_w)$, and (d_l, d_w) (shown as green dots in figure 2). This creates a coordinate system which is independent from the dimensions and the exact geometric distances of the table. For example, the center dot of the football field will always have the coordinate $(0.5d_l, 0.5d_w)$ on any football table.

The positions of the foosmen in this coordinate system can be described in several constants and variables. The constants are derived from the topology invariants, and are shown in figure 2. These constants have an unknown fixed ratio to d_l and d_w. As such, in order to accurately describe the position of the foosmen within the coordinate system, it is necessary to estimate these *topology ratios*. The variables describe the movement of the foosmen along the rods in the y-axis.

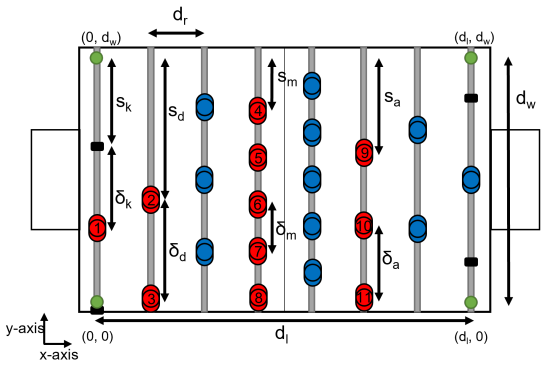


Fig. 2: Topology description of the football table.

The setup consist of two sensors: a high speed camera, and an encoder sensor in the actuators controlling the rods. Both sensors needs to be calibrated within the new world model. With regards to the camera, it is necessary to create a camera model within the world model in which a pixel position in the camera image can be converted to a position in the coordinate system. The parameters for this camera model can be automatically estimated using the topology of the football table. The encoder sensors in the actuators of the rods can be used to determine the position of the foosmen along the rods. In order to be able to estimate the parameters of the world model, it is necessary that the 11 foosmen of the controllable team can be identified in the images from the camera. Furthermore, since all decisions depend on the location of the ball within the topology, being able to identify and then localize the ball is required as well. The software needs to determine an actuator response with the rods, depending on the location of the ball. This response depends on the workspaces of each individual foosman, because for a foosman to be able to kick the ball, it needs to be able to actually reach the ball. As such, the software needs to determine the workspace of each individual foosman.

Figure 3 depicts a flowchart of the software architecture, showing the different functionalities as described earlier and the order in which they are executed. The general structure of the architecture can be divided in three parts: an initialization part, a normal operation part, and update world model part. This division is made because some functionalities will only need to be executed once, while other functionalities need to be executed continuously.

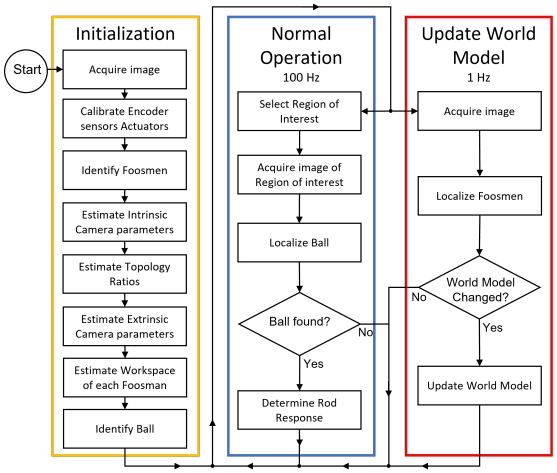


Fig. 3: Flowchart of software operations

The software starts with the *initialization* part, in which the software first needs to calibrate the encoders sensors in the actuators, and then identify the controllable foosmen. When the foosmen are identified, they can be used to estimate a complete world model, which consist of several camera parameters, the topology ratios, and the dimensions of the workspaces of each individual foosman. Lastly, with the world model complete, the ball can be identified and the software is ready to play the game. The initialization part will only run once at the start.

The *normal operation* part are the actions that are continuously executed in order to play the actual game of table football. In order to accurately track the ball, this part operates at a frequency of 100 Hz, which was determined in [5] to be minimal frequency required for this task. The normal operation part consists of selecting a correct region of interest which most likely contains the ball, acquire an image of the chosen region of interest, localize the ball in this image, and determine a response with the rods based on this information. These actions loop continuously. Using a smaller image of a chosen region of interest instead of using the entire image is necessary in order to save processing time and to be able to reach the necessary frequency of 100 Hz. The region of interest is chosen based on the last known pixel location of the ball in the image. If the ball is not visible in the chosen region of interest, the software will cycle through different regions of interest until the ball is localized again.

The *update world model* part is there to check, and if necessary correct, the world model. The world model is not necessarily static during active play. For instance the camera might move during active play, which means that certain aspects of the world model will need to be updated. In order to check if the world model has changed, the software localizes the eleven foosmen in the image and checks if these locations correspond to the predicted locations of the foosmen within the world model. If there is a mismatch, the software will update the world model so that the actual locations match the predicted locations in the world model again. The update models part is continuously executed parallel to the normal operation part, but at a lower frequency. The reason for this is the fact that checking and updating the world model is a demanding task, and if it were performed at the same frequency as the normal operation, the computer would not be able to process it in time. Furthermore, while high frequencies are necessary to track the ball, such high frequencies are not required to update the world model, as the world model is not going to change at a frequency of 100 Hz. The exact frequency at which this part operates depends on the processing power of the computer. For the current computer, a frequency of 1 Hz is chosen, which is fast enough in order to update the world model without much delay, and also not too demanding so that the normal operation part is able to run at 100 Hz.

III. Identifying Foosmen and Ball

In order to be able to identify the foosmen and the ball in the image, image segmentation and feature detection is used. Image segmentation divides the image in different segments. Feature detection is then performed on these segments in order to determine which segments correspond to the foosmen and the ball. There are many algorithms that can be used for this purpose, ranging from classical computer vision approaches such as thresholding and edge detection [15], all the way to modern AI-based approaches such as neural networks. An important consideration in this research is the fact that the software has to operate at a frequency of 100 Hz. This means that the chosen algorithms has to have a low computational cost. Furthermore, since manual calibrations are undesired, an algorithm that needs to be trained is not a right choice. Because of both these reasons, AI-based approaches are not desired [12]. Furthermore, the object detection is performed in a controlled setting, i.e. it will be an image of a football table field with a set of invariants which can be used in identification. These invariants can be used as the features in the feature detection with which the foosmen and ball are identified. Color is an obvious feature to use for object detection, as both the foosmen as the ball will have an unique and uniform color, and using color as a feature has a low computational cost. Color alone is not enough however, as using just color would result in false positives, and the foosmen and ball colors are not known at the start. So other features are necessary to identify the objects, especially the first time when the colors are not yet known.

A. Identifying the Foosmen

The initial feature used to identify the foosmen is movement. During initialization, all controllable eleven foosmen will be simultaneously moved by the software, after which background subtraction [16] can be used to identify which objects moved in the images. Since there are eleven foosmen, there should be eleven moving objects of similar color if no other movements have taken place. If more than eleven moving objects are located via background subtraction, then these extra objects will stand out because of its differing color, and thus can be excluded. After the foosmen are identified the first time, the team color is determined, which will be used for future identification. The foosmen color is estimated in HSV color space. In HSV color space the color and the intensity are separated from each other, which makes color detection more robust against variations in lighting when compared to a RGB color space, where variations in lighting produce wide variations in all three components [11].

After initialization, the foosmen are identified using color. Thresholding is performed on the image [15], creating a binary image with all pixels close to the foosmen color maximized, and all other pixels minimized. Using dilation and erosion, noise is removed and nearby pixels are grouped as *blobs* [15]. Each blob is regarded as a potential foosman. The topology of the foosmen can then be used to analyse this list of potential foosmen so as to separate the actual foosmen from false positives. Since the pose of the camera is not pre-determined and can change during play, nothing is known about the perspective effects the image can have on the topology of the foosmen. As such, it is necessary to identify the topology of the foosmen using *projective invariants*. One such invariant is the collinearity of points: If a set of points are collinear in world coordinates, than this set of points will also be collinear in the image. This aspect can be used to locate the five midfield foosmen, which will always appear as a line in the image. Another useful projective invariant is the cross-ratio [17]. The cross-ratio of a set of four collinear points will be constant despite any perspective effect in the image. Given four collinear points A, B, C, and D, their cross-ratio Cr is defined as:

$$Cr = \frac{\overline{AC} \cdot \overline{BD}}{\overline{BC} \cdot \overline{AD}} \tag{1}$$

With the bar indicating the distance between two points. Since the five midfielders are equally spaced apart by distance δ_m, the cross-ratio of four neighboring midfielders will be $\frac{4}{3}$. This cross-ratio can be used to determine which five points are most likely to correspond to the midfield rod. The three attack foosmen can be identified using the invariant of collinearity again. Unfortunately the potential attack rod cannot be verified using the cross-ratio again, as there are only three attackers. However, the angle of the potential attack rod can be compared to the angle of the earlier determined midfield rod, as both angles should be similar. When both the midfield foosmen and the attack foosmen are located, the probable positions of the defenders and keeper can be predicted. Determining which points in the remaining set of potential foosmen corresponds to the defenders and keeper is then straightforward.

B. Identifying the Ball

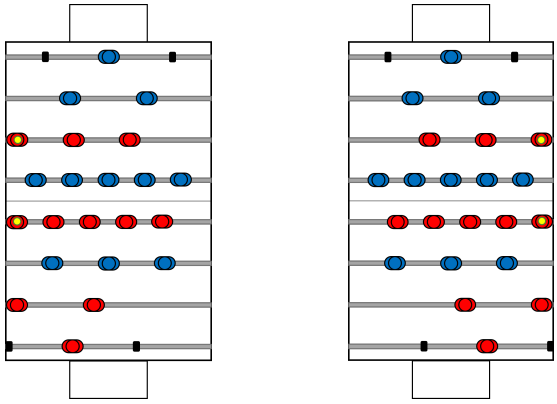
The initial features used to identify the ball are its shape, relative size, and position in the image. The shape of the ball is simply an ordinary circle shape, as long as it is not occluded by other objects in the image. While the absolute size of the ball is impossible to know, the relative size of the ball in relation to the foosmen is known (one of the topology invariants), and as such, if the foosmen are identified, an estimate of the relative size of the ball can be made. Furthermore, since the foosmen are identified, it is possible to estimate the edges of the field in the image. Any potential ball whose position is outside the edges can be excluded. K-means clustering is performed on the image, in order to reduce the color palette of the image [15]. This results in a image where edges are more clearly defined. The canny edge detection can then be performed on the image to give a set of edges [18]. From the resulting set of edges the closed edges with a circular shape are extracted. The set of circular edges is then filtered by removing the circular shapes that are in positions corresponding to known circular shapes such as the center circle, and positions which are outside the field. The set is also filtered by removing circles that are too big or too small relative to the expected size of the ball. The remaining circle should then correspond to the ball.

After the first identification, the color of the ball is determined in a similar manner as the foosmen, namely using the HSV color space and thresholding. For the rest of the game, the identification is done through this color and position. The position of a potential ball is used by checking if it's inside the edges of the field. Occlusion of the ball happens often, which makes shape and relative size an unreliable property to constantly use for ball detection. Furthermore, k-means clustering is too computationally demanding when the required frequency of 100 Hz is to be reached, and edge detection is too unreliable without this clustering beforehand.

IV. Estimating World Model

A. Calibrating the Actuator Sensors

The setup consists of two sensors: the encoder sensors in the actuators controlling the rods, and the high speed camera aimed at the field. From the data of the encoder sensors it is possible to estimate where a set of foosmen is located along the rod. Each set of foosmen on a rod has two extreme positions, namely when the foosmen are located against the two sides of the table at the ends of the rods. In the world model, each of the four controllable rods has one variable (α_k, α_d, α_m, and α_a) that describe where the set of foosmen on that rod is located in between these extreme positions. An α equal to zero corresponds to the extreme left location on the rod and an α equal to one corresponds to the extreme right location on the rod (as seen in figure 4).



(a) Image L, when all $\alpha = 0$ **(b)** Image R, when all $\alpha = 1$

Fig. 4: Layout of football table when all α's are equal to 0 and 1. Yellow dots represents locations used to create a rectangle as can be seen in figure 5

During initialization, the software will slowly move all foosmen to the left until no further change in the actuator sensors is detected anymore, which is the point where the foosmen hit the left wall. The current sensor readings are saved as the constant P_{Li}, with i corresponding to k, d, m, or a depending on the rod. All foosmen are then moved to the right until the right wall is hit, and the sensor readings are saved as the constant P_{Ri}. If the variable P_i is defined as the current sensor reading during the game, then the current α_i is defined as:

$$\alpha_i = \frac{P_i - P_{Li}}{P_{Ri} - P_{Li}} \tag{2}$$

B. Camera Model

To translate an image from the camera to usable information for the world model, a projective transformation between the 3d scene of the football table and 2d pixels in the image is necessary. Starting from the standard pinhole camera model, as is given from [6], results in the following:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{T}] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3}$$

Where $\mathbf{K}$ is the intrinsic camera matrix, which projects 3d points given in the frame of the camera to the 2d pixel coordinates in the image, and where $\mathbf{R}$ and $\mathbf{T}$ are the rotation and translation matrices respectively, which together are known as the extrinsic camera matrix and describe the coordinate conversion from the world frame to the camera frame. u and v are the pixel locations in the image in the x-axis and y-axis respectively, s is the projective transformation's scale factor, and $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$ are the 3d coordinates in the world frame (which was described in II, see figure 2 for world frame). The intrinsic matrix $\mathbf{K}$ consists of 5 unknown parameters. γ, which represents the skew between the x and y axis, is usually nonexistent in modern cameras [19], and as such, will be assumed to be 0. Furthermore, since the height and width of a pixel in modern cameras is usually equal [19], the focal lengths f_x and f_y are considered to be equal as well, simplifying those two parameters to just one parameter f. u_0 and v_0 are the x and y coordinates of the principal point of the image, which is usually in the center of the image.

Real cameras have lenses, which introduces lens distortions. The most common and severe distortion in modern cameras is radial distortion [6]. As such, radial distortion will be considered in the this paper, while other distortions are assumed to be small enough to be ignored. There are several ways to model radial distortion, the most common of which is the even-order polynomial model [20]. However, in this paper, a one order division model is used which provides a more accurate approximation than the even-order polynomial model, while needing less parameters [21]. This model is written as:

$$\mathbf{v_u} = (1 + \lambda r_d^2)^{-1}(\mathbf{v_d} - \mathbf{v_0}) + \mathbf{v_0} \tag{4}$$

Where $\mathbf{v_u} = \begin{bmatrix} u_u & v_u \end{bmatrix}^T$ are the undistorted pixel coordinates and $\mathbf{v_d}$ the distorted pixel coordinates. λ is the distortion parameter. r_d can be described as:

$$r_d = \sqrt{(u_d - u_0)^2 + (v_d - v_0)^2} \tag{5}$$

A common method to determine the parameters present in the pinhole camera model and the lens distortion model is to calibrate the camera using a checkerboard or a similar object in different poses in the image [6]. This method is based on the fact that a calibration object such as a checkerboard has a known familiar pattern. The perspective distortion of this familiar pattern can then be exploited to estimate the parameters in the camera models. This paper presents a method to determine these parameters completely automatic and based on the known features and topology of the football table, without the need for extra objects such as a checkerboard.

C. Estimating the Intrinsic Camera Parameters

The lens distortion model (4) is estimated first. To determine λ, a method is used as is presented from [8]. Here, λ can be determined by looking at the amount of distortion in straight lines. Straight lines are abundant in a football table, but due to radial distortion, all these lines will be curved in the image. By finding out the curvature of these lines, an estimation can be made for λ.

First, a canny edge detection [18] is performed on the image. The Harris corner detection [22] is then performed on the set of edges resulting from the canny edge detection. The corner detection is used to separate the edges in a set of curved lines. These curved lines can be treated as circular arcs. The radius and center coordinates of these circles can be determined by fitting a circle to the curved line. According to [8], the best circle fits are given by the Levenberg Marquardt circle fit [23]. However, this circle fit algorithm requires an initial guess for the circle parameters. As such, this paper uses the Taubin circle fit [24] to find an initial guess for a circle, which will then be refined using the Levenberg Marquardt circle fit. When the circles are determined, λ can be estimated from such a circle using the following equation as given from [8]:

$$\lambda^{-1} = u_0^2 + v_0^2 - 2c_u u_0 - 2c_v v_0 + c_u^2 + c_v^2 - c_r^2 \tag{6}$$

Where c_u and c_v are the pixel coordinates of the center of the circle, and c_r is the radius of the circle. Longer curved lines will give more accurate results for λ, as well as curved lines which are located further away from the principal point given by $\mathbf{v_0}$. As such, the resulting λ's from these lines are given a higher weight. λ's which differ substantially from the median of all λ's are excluded from the set. The remaining λ's are averaged, resulting in a final λ which is used in the model. Using a few seconds of images from the camera increases the precision of the estimation, as it is possible that a single image contains enough noise that the estimation of lambda is off. Using multiple images ensures that these noisy images can be excluded from the estimation.

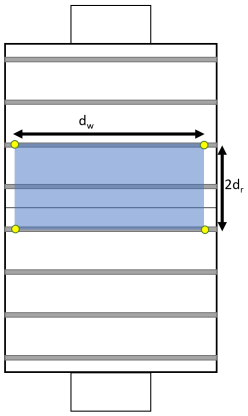
When the image can be corrected with the estimated distortion model (4), the focal length f (and by extension the intrinsic matrix $\mathbf{K}$) can be estimated. This is done using two images of the football field: one image in which all actuator values $\alpha = 0$ (called image L), and one image in which all $\alpha = 1$ (called image R). This set-up can be seen in figure 4. By combining the foosmen positions in image L and R, it is possible to connect certain foosmen locations so that a rectangle can be formed in the world frame, as shown in figure 5a. From the perspective of the camera however, this rectangle will be deformed into a certain quadrilateral, as can be seen in figure 5b. According to [7], this deformation can be used to estimate the focal length f.

The focal length f can be derived using:

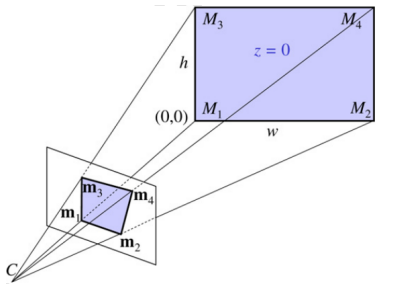
$$\mathbf{n_w}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{n_h} = 0 \tag{7}$$

In which $\mathbf{K}$ is the intrinsic matrix, and where $\mathbf{n_w}$ and $\mathbf{n_h}$ are 3 dimensional vectors that can be described as:

$$\begin{aligned} \mathbf{n_w} &= \frac{(\mathbf{m_1} \times \mathbf{m_4}) \cdot \mathbf{m_3}}{(\mathbf{m_2} \times \mathbf{m_4}) \cdot \mathbf{m_3}} \mathbf{m_2} - \mathbf{m_1} \\ \mathbf{n_h} &= \frac{(\mathbf{m_1} \times \mathbf{m_4}) \cdot \mathbf{m_2}}{(\mathbf{m_3} \times \mathbf{m_4}) \cdot \mathbf{m_2}} \mathbf{m_3} - \mathbf{m_1} \end{aligned} \tag{8}$$



(a) Created rectangle based on foosmen positions in figure 4



(b) Deformation of a rectangle in a quadrilateral due to perspective effect. Figure from [7].

Fig. 5

In which $\mathbf{m_1}$, $\mathbf{m_2}$, $\mathbf{m_3}$, and $\mathbf{m_4}$ are the pixel coordinates of the observed corners of the quadrilateral in the image (see figure 5b). The aforementioned corners are written as augmented vectors, i.e.: $\mathbf{m_1} = \begin{bmatrix} u_{m1} & v_{m1} & 1 \end{bmatrix}^T$. Using (7), the focal length f can be described as:

$$f^2 = -\frac{1}{n_{w3}n_{h3}}\Big(n_{w1}n_{h1} + n_{w2}n_{h2} - (n_{w1}n_{h3} + n_{w3}n_{h1})u_0 - (n_{w2}n_{h3} + n_{w3}n_{h2})v_0 + n_{w3}n_{h3}u_0^2 + n_{w3}n_{h3}v_0^2\Big) \tag{9}$$

Where in n_{wi} and n_{hi} the i denotes the ith component of the vectors $\mathbf{n_w}$ and $\mathbf{n_h}$. Estimating the focal length is impossible when either n_{w3} or n_{h3} is equal to zero, because (9) is not defined in that situation. This situation occurs when the camera is located directly above the football table, and aimed perpendicular to the football field. In this situation, there is no perspective effect, and as such, it is impossible to determine a focal length. This means that in this particular situation the complete camera model (3) cannot be estimated. This specific situation and the solutions required to solve it are discussed in the following sections where a known focal length is necessary.

D. Estimating the Extrinsic Camera Parameters

If the intrinsic matrix is known, then the extrinsic matrix can be found by matching specific pixel locations in the image to the corresponding 3d locations in the world coordinates. This is known as a perspective-n-point problem (PnP), where the pose of the camera is estimated by relating a set of known 3d points in the world frame to their corresponding pixel positions in the image [13]. The foosmen can be used for this purpose, giving eleven positions to match the image to the 3d scene.

As was stated earlier in section II, in the chosen coordinate system, the positions of the foosmen can be described in several constants related to the topology (see figure 2), and the four actuator variables α as defined in (2). Table I lists each foosmen in terms of these topology constants and actuator variables in the left side of this table. These coordinates are directly derived from the topology invariants. Which foosman corresponds to the which foosman number can be seen in figure 2.

TABLE I:
FOOSMEN COORDINATES

	X	Y
1	0	$\alpha_k s_k + \delta_k$
2	d_r	$\alpha_d s_d$
3	d_r	$\alpha_d s_d + \delta_d$
4	$3d_r$	$\alpha_m s_m$
5	$3d_r$	$\alpha_m s_m + \delta_m$
6	$3d_r$	$\alpha_m s_m + 2\delta_m$
7	$3d_r$	$\alpha_m s_m + 3\delta_m$
8	$3d_r$	$\alpha_m s_m + 4\delta_m$
9	$5d_r$	$\alpha_a s_a$
10	$5d_r$	$\alpha_a s_a + \delta_a$
11	$5d_r$	$\alpha_a s_a + 2\delta_a$

TABLE II:
FOOSMEN COORDINATES IN RATIOS WITH d_w

	X (in d_w)	Y (in d_w)
1	0	$\alpha_k {}^{s_k}/_{d_w} + {}^{\delta_k}/_{d_w}$
2	$\frac{1}{7}{}^{d_l}/_{d_w}$	$\alpha_d {}^{s_d}/_{d_w}$
3	$\frac{1}{7}{}^{d_l}/_{d_w}$	$\alpha_d {}^{s_d}/_{d_w} + {}^{\delta_d}/_{d_w}$
4	$\frac{3}{7}{}^{d_l}/_{d_w}$	$\alpha_m {}^{s_m}/_{d_w}$
5	$\frac{3}{7}{}^{d_l}/_{d_w}$	$\alpha_m {}^{s_m}/_{d_w} + {}^{\delta_m}/_{d_w}$
6	$\frac{3}{7}{}^{d_l}/_{d_w}$	$\alpha_m {}^{s_m}/_{d_w} + 2{}^{\delta_m}/_{d_w}$
7	$\frac{3}{7}{}^{d_l}/_{d_w}$	$\alpha_m {}^{s_m}/_{d_w} + 3{}^{\delta_m}/_{d_w}$
8	$\frac{3}{7}{}^{d_l}/_{d_w}$	$\alpha_m {}^{s_m}/_{d_w} + 4{}^{\delta_m}/_{d_w}$
9	$\frac{5}{7}{}^{d_l}/_{d_w}$	$\alpha_a {}^{s_a}/_{d_w}$
10	$\frac{5}{7}{}^{d_l}/_{d_w}$	$\alpha_a {}^{s_a}/_{d_w} + {}^{\delta_a}/_{d_w}$
11	$\frac{5}{7}{}^{d_l}/_{d_w}$	$\alpha_a {}^{s_a}/_{d_w} + 2{}^{\delta_a}/_{d_w}$

The X coordinates of the foosmen can be described using the distance between two neighboring rods d_r. The Y coordinates of the foosmen are described using the the spacing between the foosmen δ_i, the length of the stroke of the foosmen s_i, and the value of the actuators controlling the foosmen α_i, with i corresponding to a rod. The Z coordinates of the foosmen are all equal to zero, as all foosmen are located in the same plane. As such, the Z coordinates are not shown in table I.

Since the real geometric values of these constants are not known, it is not possible to use these coordinates directly in this way to solve the PnP problem. However, the PnP problem is also solvable when these coordinates are known in terms of one chosen topology constant. The logical chose for this constant is either d_l or d_w, as these are the constants corresponding to the x and y-axis respectively in the world model (see section II). d_w is chosen here as it is more convenient when calculating the topology ratios than d_l, although d_l would have worked just as well. If the ratios between all topology constants and d_w can be estimated, then it is possible to describe all foosmen positions as dimensionless coordinates in terms of d_w. Table II shows the same foosmen coordinates as table I, but now redefined as ratios with d_w. This results in the set of topology ratios that now need to be estimated, after which these coordinates can be used to solve the PnP problem. To solve the PnP problem, the SQPnP algorithm as described in [14] is used, which is an algorithm that always determines the global minima of the PnP problem and consistently achieves results at a low computational cost.

E. Estimating the Topology Ratios

The topology ratios in the y-axis can be estimated as follows. Take any two neighboring foosmen on the same rod,

called foosman A and foosman B. During the calibration of the actuator sensors as described in subsection IV-A, all foosmen are located at the extreme left position during which all α's are zero (see figure 4a). The position of foosmen A and B in world coordinates can be expressed as A_L and B_L, and in pixel positions in the image as a_L and b_L. All foosmen are then moved to the extreme right position during which all α's are one (see figure 4b). The position of foosmen A and B in world coordinates can be expressed as A_R and B_R, and in pixel positions in the image as a_R and b_R. The four points A_L, B_L, A_R, and B_R are collinear, and as such, the cross-ratio can be used to relate the four world points to the four image points [17]. The cross-ratio is a projective invariant and can be used to relate the points as:

$$\frac{\overline{A_L A_R} \cdot \overline{B_L B_R}}{\overline{B_L A_R} \cdot \overline{A_L B_R}} = \frac{\overline{a_L a_R} \cdot \overline{b_L b_R}}{\overline{b_L a_R} \cdot \overline{a_L b_R}} = Cr \tag{10}$$

In which the bar above two points indicates the distance between the two points, and Cr is the invariant cross-ratio. The right side of (10) is known, as it can be directly inferred from pixel measurements. Filling in the left side of (10) with the stroke length and foosmen spacing constants results in:

$$\frac{s_i \cdot s_i}{(s_i - \delta_i) \cdot (s_i + \delta_i)} = Cr \tag{11}$$

In which i is the rod corresponding to the chosen foosmen A and B. Since s_i is always larger than δ_i (a topology invariant), Cr is always a number larger than 1. As can be derived from the topology invariants and figure 2, d_w can be described as:

$$d_w = s_i + (n_i - 1)\delta_i \tag{12}$$

Where n_i is the amount of foosmen on that specific rod. Combine (11) and (12) to get the ratios ${}^{s_i}/_{d_w}$ and ${}^{\delta_i}/_{d_w}$ as in (13) and (14) respectively:

$$\frac{s_i}{d_w} = \left(1 + (n_i - 1)\sqrt{\frac{Cr - 1}{Cr}}\right)^{-1} \tag{13}$$

$$\frac{\delta_i}{d_w} = \left(\sqrt{\frac{Cr}{Cr - 1}} + (n_i - 1)\right)^{-1} \tag{14}$$

This method can be repeated on any set of neighboring foosmen to estimate all necessary ratios. The method does not work on the ratios ${}^{s_k}/_{d_w}$ and ${}^{\delta_k}/_{d_w}$ however, as the keeper has no neighboring foosmen. These ratios can be derived later however. By solving the PnP problem using just ten foosmen instead of eleven (by excluding the keeper foosman), it is possible to derive the Y coordinate of the keeper using (3), which can then be used to derive the ratios by using table II.

The rod distance d_r and the topology length d_l are related to each other by a fixed ratio:

$$d_l = 7d_r \tag{15}$$

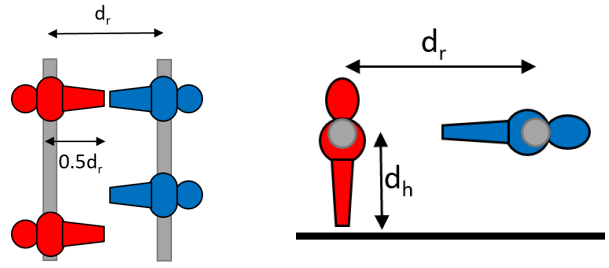
This follows from one of the topology invariants, which states that a football table consists of 8 equally spaced rods. The ratio between d_l and d_w can be estimated using the earlier created quadrilateral shown in figure 5b, which was used to estimate the focal length. [7] suggest that the aspect ratio of this deformed rectangle in figure 5a can be determined using:

$$(\frac{2d_r}{d_w})^2 = \frac{\mathbf{n_w}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{n_w}}{\mathbf{n_h}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{n_h}} \tag{16}$$

In which $\mathbf{n_w}$ and $\mathbf{n_h}$ are defined by (8). The fraction ${}^{2d_r}/_{d_w}$ is the aspect ratio of the deformed rectangle. This determined aspect ratio can then be used to estimate the ratio ${}^{d_l}/_{d_w}$ via (15). If the situation occurs where (9) is undefined and the focal length is unknown, the quadrilateral in the image from figure 5b will not be deformed, and as such will be visible as an rectangle in the image. This means that the aspect ratio in the image will be equal to the aspect ratio in world coordinates. As such, the constant d_l can still be related to d_w by simply estimating the aspect ratio of rectangle in the image.

F. Using the estimated model to locate the ball

The now complete camera model from (3) can be used to project a 3d point in the world frame to the image. The inverse operation is necessary however. This is impossible, since in this situation only u and v are known in the model while X, Y, Z, and s are unknown, which means the model is undetermined. However, an assumption can be made, which does make this solvable. By assuming that the ball will always be located on the field (which for all intents and purposes it will be), the position of the ball in the z-axis is fixed at a certain constant d_h, which is the distance between the rods and the field (see figure 6b). By assuming a fixed position of the ball in the z-axis, only the position of the ball in the other two axis remain unknown. By combining the fixed distance in the z-axis with the pixel location of the ball in the image, enough variables are known in order to use the camera model (3).



(a) Layout when foosmen are turned 90°

(b) Sideview of foosmen

Fig. 6

The constant d_h can be related to the other constants via d_r. When the foosmen on two neighboring rods are angled parallel to the x-axis (90°turn), the feet of two opposing foosmen can almost touch (see figure 6a), which was one of the topology invariant. This means that the length of a foosman from feet to shoulders (where it is connected to the rod) will be equal to $0.5d_r$. Turning this foosman 90°to the normal position, will show that the foosman will slightly hover above the field. This

means that the field is located a foosman length below the rods (see figure 6b). As such d_h can be described as:

$$d_h = 0.5 d_r \tag{17}$$

The camera model (3) can be changed as follows to create a new model from which the position of the ball can be estimated. First, the pixel coordinates of the ball in the image, u_b and v_b, are converted to normalized camera coordinates x'_b and y'_b respectively, using (3). This results in:

$$\begin{aligned} x'_b &= \frac{X_c}{Z_c} = \frac{u_b - u_0}{f} \\ y'_b &= \frac{Y_c}{Z_c} = \frac{v_b - v_0}{f} \end{aligned} \tag{18}$$

With X_c, Y_c, and Z_c as the coordinates of the ball in camera coordinates. Using these normalized camera coordinates, (3) can be rewritten as:

$$Z_c \begin{bmatrix} x'_b \\ y'_b \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} \tag{19}$$

Where X_b, Y_b, and Z_b describe the coordinates of the ball in world coordinates in terms of d_w. The value Z_b is fixed as $Z_b = d_h$, while X_b, Y_b, and Z_c are unknown. The matrix terms r_{ij} are components of the rotation matrix $\mathbf{R}$ and t_x, t_y, and t_z are components of the translation vector $\mathbf{T}$. Rearranging (19) and placing all unknowns in one vector, results in:

$$\begin{bmatrix} r_{13} d_h + t_x \\ r_{23} d_h + t_y \\ r_{33} d_h + t_z \end{bmatrix} = \begin{bmatrix} x'_b & -r_{11} & -r_{12} \\ y'_b & -r_{21} & -r_{22} \\ 1 & -r_{31} & -r_{32} \end{bmatrix} \begin{bmatrix} Z_c \\ X_b \\ Y_b \end{bmatrix} \tag{20}$$

Which can be written in the form:

$$\mathbf{h} = \mathbf{B} \begin{bmatrix} Z_c \\ X_b \\ Y_b \end{bmatrix} \tag{21}$$

By taking the inverse of $\mathbf{B}$, (21) can be used to determine the coordinates of the ball in the world frame. This method can not be used if $\mathbf{B}$ is singular. However, this is highly unlikely to happen, but if it were to happen, an easy solution would be to simply use a neighboring pixel for u_b and/or v_b in (18). This would still result in an accurate estimation of the location of the ball (one pixel difference would not make a significant difference), while creating a non-singular matrix $\mathbf{B}$.

If the situation occurs where the focal length is not known because (9) was undefined, the location of the ball can still be estimated. In this unique situation, the camera is aimed exactly perpendicular to the field. As a result, the rotation matrix $\mathbf{R}$ will take the form of:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{22}$$

In which θ is the angle of rotation of the field with respect to the camera, which can be estimated by using the known topology of the foosmen (by looking at the angle of the rods and where certain foosmen are located in the image). The t_x and t_y components from the translation matrix $\mathbf{T}$, can also be estimated using the topology. The component t_z disappears in this situation, as (19) simplifies to:

$$Z_c \begin{bmatrix} x'_b \\ y'_b \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \end{bmatrix} \begin{bmatrix} X_b \\ Y_b \\ 1 \end{bmatrix} \tag{23}$$

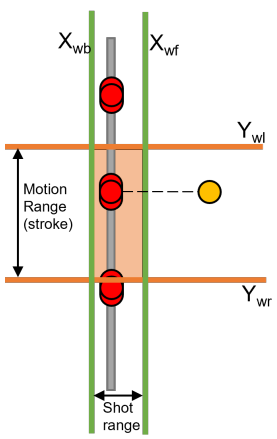
The normalized image coordinates x'_b and y'_b cannot be directly estimated without the focal length. Furthermore, Z_c is still unknown. From (18), the camera coordinates can be described as:

$$\begin{aligned} X_c &= \frac{Z_c}{f}(u_b - u_0) \\ Y_c &= \frac{Z_c}{f}(v_b - v_0) \end{aligned} \tag{24}$$

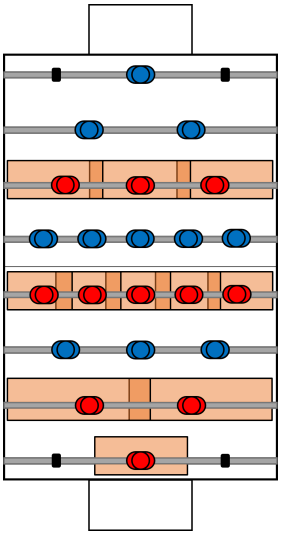
The fraction Z_c/f can be estimated by comparing the distances between foosmen in world coordinates to the same distances between foosmen in pixel coordinates. The ratio between these distances in world coordinates and in pixel coordinates will be the same for every distance, and will in fact be Z_c/f. When Z_c/f, θ, t_x, and t_y have been determined, (23) and (24) can be used to estimate the position of the ball in world coordinates.

G. Estimating the workspaces of the foosmen

The workspace of an individual foosman is defined as the range of points in the world coordinates where a foosman is able to reach and kick the ball forward towards to goal of the opponent. Knowledge about the workspaces of each foosman is necessary in order to make a decision about which foosman to move. The workspace of a foosman is defined using four lines: two longitudinal lines which are parallel to the x-axis and have a Y value Y_{wl} and Y_{wr}, and two lateral lines which are parallel to the y-axis and have a X value X_{wf} and X_{wb}. The longitudinal lines represent the edges of the stroke along the rod of a specific foosman. If the ball is located in between these two lines, then it is possible to align that foosman with the ball in order to block or kick the ball if the ball crosses the path of the foosman. The lateral lines represent the edges of the range of the kick of a foosman. If the ball is located in between these two lines as well as the longitudinal lines, then it is possible for that foosman to kick the ball forward. If the ball is located behind the lateral line with value X_{wb}, then this foosman will turn to an angle such that the ball will not be blocked if it is shot forward from behind. The workspaces of neighboring foosmen can overlap. When the ball is in more than one workspace, either foosmen can be chosen. In this situation, the foosman that has the current shortest distance to the ball will be chosen. Figure 7a shows how the workspace is defined for the middle attacker



(a) The four lines defining the workspace of the middle attacker foosman. The two longitudinal lines are used to position the foosman with respect to the ball. The lateral lines are used to determine where the ball can be kicked.



(b) Workspace of each red foosman shown as a transparent orange rectangle. Some workspaces overlap.

Fig. 7

foosman, while figure 7b show the workspaces of all foosmen.

The parameters Y_{wl} and Y_{wr} for a foosman can be defined by using table II. The left edge Y_{wl} of the workspace of each foosman is defined as the Y coordinate of that foosman when α is 0. The right edge Y_{wr} of the workspace of each foosman is defined as the Y coordinate of that foosman when α is 1. The parameters X_{wf} and X_{wb} for a foosman can be estimated in a similar way. The front edge X_{wf} of the workspace of the each foosman is defined as the X coordinate of that foosman plus d_{f1} as estimated in (25). The back edge X_{wb} of the workspace of the each foosman is defined as the X coordinate of that foosman minus d_{f2} as estimated in (25).

$$\begin{aligned} d_{f1} &= \sqrt{d_h^2 - (d_h - r_b)^2} \\ d_{f2} &= d_{f1} - r_b \end{aligned} \tag{25}$$

Where r_b is defined as the estimated radius of the ball in world coordinates. Figure 8 shows how d_{f1} can be estimated using d_h and r_b and the Pythagorean theorem.

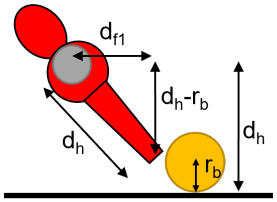


Fig. 8: The figure shows the maximum distance d_{f1} at which the ball can still be kicked by the foosman.

V. Experiments and Results

In order to test the hypothesis that the presented method is robust against variations, a series of experiments are conducted under changing parameters. These changing parameters are different camera angles, as well as different football tables with different dimensions and colors. The experiments are divided in four subsections. In the first three subsections 40 measurements are conducted on EUTAFT. In these 40 measurements, four different camera poses are tested (10 measurements per camera pose). The four chosen camera poses can be seen in figure 9. The first subsection focuses on the accuracy of the estimated camera model, the second subsection focuses on the accuracy of the estimated topology ratios, and the third subsection focuses on the accuracy of the ball tracking and actuator responses. The last subsection will focus on additional experiments that were performed on three additional football tables with different dimensions.

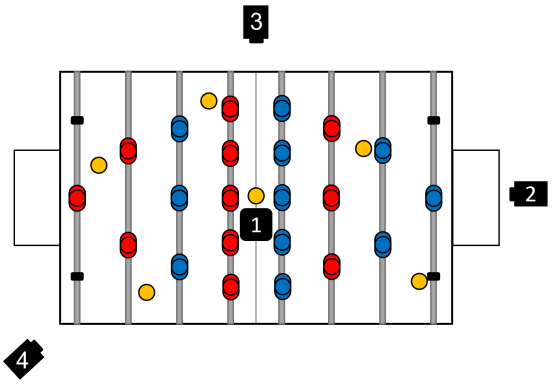


Fig. 9: The four camera positions and the six ball locations used in the experiments

A. Validation of estimated camera model

The estimated camera model can be validated by comparing the estimated camera model to a camera model found using a more traditional method, namely a camera model found using the checkerboard method by [6]. The difference between the intrinsic camera matrices should be minimal between the two camera models. The principal point (u_0, v_0) and the skew γ in the intrinsic camera matrix were assumed to be equal to the center of the image (which for the used camera would be equal to $(540, 720)$) and zero respectively. Furthermore, the focal length f_x and f_y were assumed to be equal. When these assumptions are compared to a camera model estimated using the checkerboard method (see 26), it can be seen that these assumptions hold. The estimated principal point differs by around 10 pixels in the u direction compared to the assumption, but [25] suggests that such an offset is not significant in the estimation of principal points.

$$\mathbf{K_{checkerboard}} = \begin{bmatrix} 1152 & 0 & 530 \\ 0 & 1152 & 721 \\ 0 & 0 & 1 \end{bmatrix} \tag{26}$$

Table III shows the mean μ and standard deviation σ of the focal length f and the distortion parameter λ of the ten experiments per camera pose. Camera pose 1 has no estimated focal length, as it is not possible to estimate an accurate focal length in this camera pose, as was discussed in subsection IV-C. The estimated focal lengths f can be

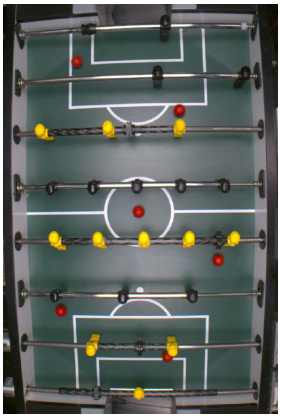
compared to the focal length found using the checkerboard method. The estimated distortion parameter is more difficult to compare to the found distortion parameters from the standard checkerboard method, as the checkerboard method uses a different lens distortion model consisting of multiple distortion parameters.

TABLE III:
CAMERA MODEL ESTIMATIONS

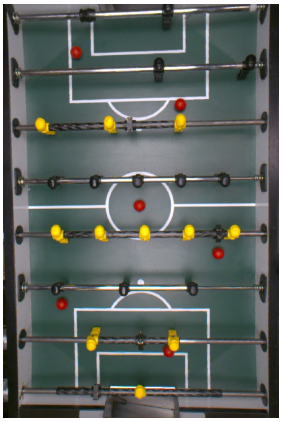
$\mu \pm \sigma$	Cam 1	Cam 2	Cam 3	Cam 4
f	-	1142 ± 38	2003 ± 363	1174 ± 32
$\lambda(10^{-9})$	153 ± 1.3	151 ± 4.2	157 ± 1.4	150 ± 1.9

The intrinsic matrix as derived from the checkerboard method is assumed as the ground truth to which the estimated focal length is compared. [25] suggests that an offset between 0 and 25 in the parameter value of f from the true value is common in the checkerboard method. As such, this paper considers a difference of less than 25 between the checkerboard focal length and the estimated focal length to be an accurate estimation. The focal length estimations for camera pose 2 and 4 are close to focal length as derived from the checkerboard method, and fall within the limit of 25. Furthermore, the variation between the ten experiments for each of these camera poses is low (the standard deviation is just 38 and 32). The estimated focal length of camera pose 3 is not close to the checkerboard focal length at all, and the standard deviation is high, meaning that there are large differences between the estimations in the experiments from this pose. In order to determine if the errors in the estimated focal lengths for camera 3 are too large for this estimation to be useful, the experiments in subsections V-B and V-C are performed twice, one time where the estimated focal length is used, and one time where the focal length is not estimated but instead fixed at the determined ground truth of 1152. The differences in these results will indicate if the error in the focal length estimation is problematic or not.

As can be derived from (4), the distortion parameter λ starts to have an effect from values of at least 2e-9. Values lower than 2e-9 would only have sub-pixel effects, meaning no real distortion correction would happen. While it is difficult to verify the validity of the estimated distortion parameters, as they cannot be compared to the checkerboard estimation, all estimated distortion parameters are in a very similar range. Using the value 2e-9 as the minimum threshold, it can be seen that the standard deviation is very small in all four camera poses. This is an indication that the estimated distortion parameters are a correct estimation, although the chance exist that there is a systematic bias present in all estimations. Figure 10a shows the image without any correction applied to it, while figure 10b shows an image where the lens distortion correction is applied with the parameter equal to 153e-9, which is the mean of all 40 estimated distortion parameters. Judging by eyesight, figure 10a clearly has barrel distortion in the image, which seem to be completely resolved in figure 10b, which indicates that there does not seem to be a systematic bias present in the estimation.



(a) Image of the table when no distortion correction is applied.



(b) Image of the table when distortion correction of 153e-9 has been applied.

Fig. 10

B. Validation of estimated Topology ratios

To validate the estimated world model, the estimated topology ratios are compared to the actual topology ratios, which were measured by hand at millimeter scale. Table IV shows the list of the mean μ and standard deviation σ of all topology ratios estimated during all experiments, with the exception of the ratio d_l/d_w, which is presented separately in table V. The estimations are not presented separately per camera pose, as there was no significant difference between the estimations between different camera poses. Since these estimations do not depend on the focal length, there was no difference between the experiments with an estimated focal length or a fixed focal length, and as such, these experiments are also not presented separately.

TABLE IV:
TOPOLOGY RATIO ESTIMATIONS OF EUTAFT

	s_k/d_w	s_d/d_w	s_m/d_w	s_a/d_w	δ_k/d_w	δ_d/d_w	δ_m/d_w	δ_a/d_w
real(10^{-3})	277	648	247	416	362	352	188	292
$\mu(10^{-3})$	281	653	245	419	360	347	189	291
$\sigma(10^{-3})$	0.5	1.3	0.4	0.7	0.2	1.3	0.1	0.4

The differences between the measured topology ratios and the estimated topology ratios are small. The largest offset between the measured ratio values and the estimated ratio values is just 0.006. The real metric value of d_w is 64 cm on EUTAFT. This means that a difference of 0.006 between the measured ratio values and the estimated ratio values would result in offset of 0.384 cm. Considering the fact that the width of the foosmen feet is 2 cm and the ball has a diameter of 3 cm, an offset of 0.384 cm is expected to be no problem. The standard deviation is low, indicating that there are no large differences between the experiments.

The estimations for the last topology ratio, namely d_l/d_w, show more varying results, which is why this ratio is presented separately in table V. Here μ_1 and σ_1 indicate the results where the focal length is estimated, while μ_2 and σ_2 indicate the results where the focal length was fixed at 1152. The real ratio is equal to 1.64.

TABLE V:
THE MEAN AND STANDARD DEVIATION PER CAMERA POSE FOR THE d_l/d_w ESTIMATION

real: 1.64	Cam 1	Cam 2	Cam 3	Cam 4
$\mu_1 \pm \sigma_1$	1.63 ± 0.003	1.59 ± 0.017	1.34 ± 0.126	1.65 ± 0.008
$\mu_2 \pm \sigma_2$	1.63 ± 0.002	1.61 ± 0.005	1.64 ± 0.006	1.65 ± 0.003

Especially noticeable is the difference between the real ratio and the estimated ratio in camera pose 3 when using the estimated focal lengths. This is a clear consequence from the inaccurate estimation of the focal length in camera pose 3 in section V-A, as this offset is not present when a fixed focal length is used.

C. Accuracy of ball tracking and actuator response

The main goal of EUTAFT is to accurately track the ball and formulate a correct response with the foosmen. As such, EUTAFT should be able to give a correct estimation of the location of the ball in world coordinates and give the right actuator values α for each rod such that a foosman is placed in front of the ball. An estimation for the ball location and actuator value is considered correct when the end result is that a foosman will be able to hit the ball straight ahead. The ball has a diameter of 3 cm, and the feet of the foosmen are 2 cm wide on the EUTAFT table. This means that an absolute error of 1 cm or lower results in the ball being hit straight ahead, an absolute error between 1 cm and 2.5 cm results in the ball being hit diagonally, and an absolute error greater than 2.5 cm results in the foosman missing the ball. As such, the end result should preferably be a rod response with an absolute error below 1 cm, and at the very least a rod response with an absolute error below 2.5 cm. While the presented world model in this paper is based on relative coordinates, the conversion back to geometric values is useful in this experiment to test the accuracy of EUTAFT.

In these experiments, the ball is placed in six known locations (as shown in figure 9). Figure 11 show the six actual ball positions (shown in black) as well as the corresponding estimations per ball position in the world coordinates (shown in color, with each color corresponding to a specific camera pose). Figure 11 show the estimations when the estimated focal lengths are used. The actual ball locations are measured by hand and then converted to world coordinates using the measured topology ratios (which were measured by hand in section V-B).

The ball estimations for camera pose 1 and 2 are very close to the real ball location. Camera pose 4 shows slightly more offset when compared to camera pose 1 and 2. Camera pose 3 shows a lot more offset and variations, which is a consequence of the incorrect estimation of the focal length. The severity of the error in ball tracking with an incorrect focal length estimation increases the farther away the balls are from the camera, which can clearly be seen when looking at the two balls lower in the y-axis which are located furthest away from camera 3.

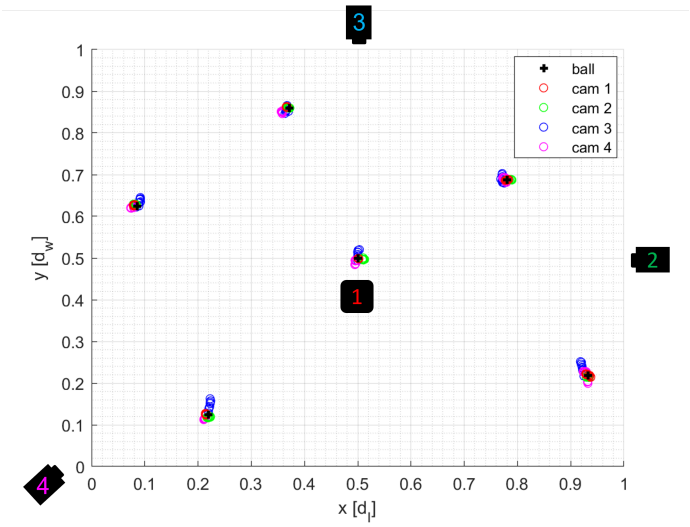


Fig. 11: Six ball positions and the corresponding estimations. The black **x**'s mark the actual ball locations, the colored **+**'s mark the corresponding estimations, where red is cam pose 1, green is cam pose 2, blue is cam pose 3, and magenta is cam pose 4.

To test the accuracy of the rod response, the desired rod actuator values are calculated using the manually measured topology ratios and ball locations. The absolute error between the desired actuator values and the chosen actuator values is converted to centimeters and plotted in figure 12 and is shown separately for each of the four camera poses.

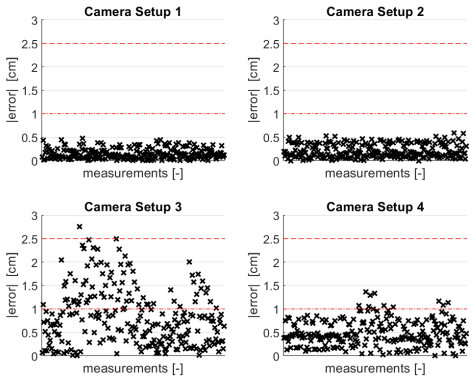


Fig. 12: Shows the absolute error of actuator values in centimeters. Error is defined as the difference between the desired actuator value and the actual actuator value. The red lines indicate the two set limits.

Table VI shows the mean and standard deviation of the absolute alignment error per camera pose, with μ_1 and σ_1 indicating the absolute error where the focal length is estimated, and μ_2 and σ_2 indicating the absolute error where the focal length was fixed at 1152.

TABLE VI:
THE MEAN AND STANDARD DEVIATION OF THE ABSOLUTE ALIGNMENT ERROR OF ALL RODS IN CENTIMETER PER CAMERA POSE

	Cam 1	Cam 2	Cam 3	Cam 4
$\mu_1 \pm \sigma_1$	0.15 ± 0.11	0.21 ± 0.14	0.82 ± 0.61	0.48 ± 0.27
$\mu_2 \pm \sigma_2$	0.14 ± 0.11	0.22 ± 0.15	0.29 ± 0.23	0.53 ± 0.34

Camera pose 1 and 2 show good results, with every single error being well below the desired value of 1 cm, and with most even below 0.5 cm. Camera pose 4 has some outliers, which causes the average error to rise and the standard deviation to be increased. However, most of these errors are still below the desired 1 cm line. This means that, in these three camera poses, all kicks will hit the ball, with all shots in camera pose 1 and 2 being straight ahead, while some shots in camera pose 4 might be accidental diagonal shots. Camera pose 3 shows substantially higher errors. When the results of camera pose 3 in table VI are compared, it can be seen that the average error and the standard deviation lower significantly when a fixed focal length is used, which indicates that the incorrect focal length estimation is the main reason for the higher rod alignment error. However, even in camera pose 3, most errors are still below the desired 1 cm limit, and except for one single measurement, all errors are below the maximum error limit of 2.5 cm. This means that EUTAFT is still accurate enough to be able to play a game of table football, even if the focal length estimation is not perfect.

D. Testing the universality of the world model

An important aspect of the new world model is the intention that this world model is robust against varying football table dimensions, as long as these other football tables follow the same set of invariants. To test if this is the case, the experiments from subsection V-B and V-C are repeated on three different football tables, which all have different dimensions and colors. While these football tables are not automated like EUTAFT, a person can stand in for the actuators of the rod.

On all three football tables, 20 measurements are performed using the same 4 camera poses as shown in figure 9 (so 5 measurements per camera pose). In this experiment, a fixed focal length of 1152 is used instead of the estimated focal length. This is done in order to focus this experiment on testing if the world model can perform on other tables. The three football tables are referred to as Football table A, B, and C in these experiments. Table VII show the real topology ratios of all three football tables, as well as the means μ_A, μ_B, μ_C and standard deviations σ_A, σ_B, σ_C of the estimated ratios.

TABLE VII:
TOPOLOGY RATIO ESTIMATIONS FOR THE NON-EUTAFT TABLES

	s_k/d_w	s_d/d_w	s_m/d_w	s_a/d_w	δ_k/d_w	δ_d/d_w	δ_m/d_w	δ_a/d_w	d_l/d_w
real A(10^{-3})	331	612	207	314	339	388	198	339	1736
μ_A(10^{-3})	334	610	213	311	333	390	197	345	1723
σ_A(10^{-3})	1.5	2.7	0.9	1.4	0.7	2.7	0.2	0.7	20.3
real B(10^{-3})	397	576	186	371	302	424	203	314	1751
μ_B(10^{-3})	395	580	187	380	303	420	203	310	1750
σ_A(10^{-3})	3.0	4.3	1.4	2.8	1.5	4.3	0.4	1.4	40.6
real C(10^{-3})	341	610	228	325	325	382	195	341	1707
μ_C(10^{-3})	341	606	226	323	330	394	194	338	1703
σ_A(10^{-3})	2.4	4.3	1.6	2.3	1.2	4.3	0.4	1.2	15.9

For all three football tables the difference between the real topology ratios and the estimated topology ratios is small. These results are of the same precision as the topology ratio estimation on EUTAFT. The offsets between the real and estimated ratios are small enough that no problems are expected for the accuracy. The standard deviation is low, indicating that there are no large differences between the measurements.

During the 20 measurements on each football table, the ball is placed in four known positions. The ball positions are estimated, and an actuator response is decided based on those results, similar as was done in subsection V-C. Table VIII shows the mean μ and standard deviation σ of the absolute alignment error per camera pose for all three football tables A, B, and C.

TABLE VIII:
THE MEAN AND STANDARD DEVIATION OF THE ABSOLUTE ALIGNMENT ERROR OF ALL RODS IN CENTIMETER PER CAMERA POSE FOR THE NON-EUTAFT TABLES

	Cam 1	Cam 2	Cam 3	Cam 4
$\mu_A \pm \sigma_A$	0.66 ± 0.45	0.40 ± 0.60	0.31 ± 0.24	0.43 ± 0.28
$\mu_B \pm \sigma_B$	0.53 ± 0.38	0.41 ± 0.27	0.46 ± 0.35	0.44 ± 0.31
$\mu_C \pm \sigma_C$	0.31 ± 0.21	0.32 ± 0.20	0.35 ± 0.22	0.41 ± 0.31

The results show that the presented method performs within the set limits, and is as such accurate enough. This demonstrates that the approach is robust against the variations these different tables have.

VI. DISCUSSION

It was hypothesised that EUTAFT could be made robust against variations by replacing the geometric based world model by a new world model based on a set of invariants. The results show that this is indeed the case. Experiments are performed using different camera setups and on multiple football tables with differing dimensions and colors. The software is able to give estimations of the camera parameters and topology ratios. These estimations are accurate enough to localize the ball and position the foosmen within the error limits such that the foosmen are able to block and kick the ball. Most balls will be hit straight ahead as most actuator errors are within the desired error limit of 1 cm. Virtually all balls will always be hit as almost no actuator error exceeds the maximum error limit of 2.5 cm. A video of EUTAFT in action can be found here.

The estimation of the focal length can be problematic, as was seen during the estimation in camera pose 3. This has consequences for the accuracy of the estimation of the topology ratio d_l/d_w and the localization of the ball. The incorrect focal length estimation for camera pose 3 suggests that this pose does not cause enough perspective deformation from which an accurate estimation for the focal length can be made. When perspective deformation is low, the terms n_{w3} and n_{h3} in (9) get closer to zero, which causes this estimation to become less accurate. Even though the performance is still within the maximum set error limits with an imprecise focal length, it might be better to simply manually calibrate the

focal length of the camera beforehand, and then fix the focal length at this value in order to minimize the error. Such a manual calibration would only need to be performed once for a specific camera. When comparing these results to the measurements from the EUTAFT football table, it can be seen that the accuracy is slightly lower on the non-EUTAFT tables, which is caused by the more prevalent outliers on these tables in comparison to EUTAFT. These outliers are most likely a result of the more difficult identification of the foosmen and ball on these three tables. EUTAFT has an active lighting system which provides bright and evenly distributed lighting directly above the table, while the other three tables depend on the surrounding light. While care was taken to make sure that the identification of the foosmen and ball is independent of lighting conditions, in practice the less optimal lighting does affect performance.

VII. Conclusion

A new world model is created which is not based on geometric distances between objects, but on pre-known and invariant relations between the objects in the workspace of the robot (which was referred to as the topology). This world model was implemented on EUTAFT, an automated football table. Earlier software on EUTAFT used a world model that used manual calibrations, 'hard-coded' geometric distances, and a fixed camera position, which made this world model not robust against variations, and as such limited the world model to one specific football table and set of sensors. The new world model requires no manual calibrations and is robust against variations such as changing camera poses and differing football tables dimensions and color, as long as the setup corresponds to a predetermined set of invariants.

In the future, improvements could be made on the estimation of the focal length of the camera. This estimation had significant offsets when the perspective effect was not large enough. This decreased the performance of EUTAFT, although performance was still good enough that this was not too problematic. Improvements can also be made on the identification of the foosmen and the ball. Lower lighting conditions reduce the performance of the new world model, as the software can have difficulties to spot the foosmen and ball in these situations. Furthermore, an assumption is made that the foosmen and ball are uniformly colored. The identification aspect of the software could be extended so that it would be possible to identify the foosmen and ball even when these are not uniformly colored. This would decrease the set of invariants, and as such, increase the universality of the new world model. Due to the accuracy of the new world model on EUTAFT, it is possible to extent the software in the future with more complex shots and strategies, such as aimed shots, or perhaps even passes to other foosmen. When implementing more complex strategies, it would be very useful if the software is able to also detect the foosmen of the opponent.

References

[1] M. T. S. W. T. Chau, J. Then and S. Cheng, "Robotic foosball table." University of Adelaide, 2007.

[2] E. T. M. Aeberhard, S. Connelly and N. Walker, "Foosball robot." Georgia Institute of Technology, 2007. [Online]. Available: http://www.eskibars.com/projects/foosball_robot/

[3] T. Weigel and B. Nebel, "Kiro – an autonomous table soccer player," vol. 2752, 11 2003, pp. 384–392.

[4] R. Janssen, J. Best, M. Molengraft, and M. Steinbuch, "The design of a semi-automated football table," 09 2010, pp. 89–94.

[5] R. Janssen, J. Best, and M. Molengraft, "Real-time ball tracking in a semi-automated foosball table," 06 2009, pp. 128–139.

[6] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[7] L.-W. H. Z. Zhang, "Whiteboard scanning and image enhancement," *Digital Signal Processing*, vol. 17, no. 2, pp. 420–424, Mar. 2007. [Online]. Available: https://www.microsoft.com/en-us/research/uploads/prod/2016/11/Digital-Signal-Processing.pdf

[8] A. Wang, T. Qiu, and L.-T. Shao, "A simple method of radial distortion correction with centre of distortion estimation," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 165–172, 11 2009. [Online]. Available: https://www.researchgate.net/publication/220146291_A_Simple_Method_of_Radial_Distortion_Correction_with_Centre_of_Distortion_Estimation

[9] S. Brahmbhatt, *Basic Machine Learning and Object Detection Based on Keypoints*. Berkeley, CA: Apress, 2013, pp. 119–153. [Online]. Available: https://doi.org/10.1007/978-1-4302-6080-6_8

[10] V. Chari and A. Veeraraghavan, *Lens Distortion, Radial Distortion*. Boston, MA: Springer US, 2014, pp. 443–445. [Online]. Available: https://doi.org/10.1007/978-0-387-31439-6_479

[11] R. Ramanath and M. S. Drew, *Color Spaces*. Boston, MA: Springer US, 2014, pp. 123–132. [Online]. Available: https://doi.org/10.1007/978-0-387-31439-6_452

[12] "Advances in computer vision," *Advances in Intelligent Systems and Computing*, 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-17795-9

[13] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, jun 1981. [Online]. Available: https://doi.org/10.1145/358669.358692

[14] G. Terzakis and M. I. A. Lourakis, "A consistently fast and globally optimal solution to the perspective-n-point problem." Springer International Publishing, 2020.

[15] S. Prabu and J. Gnanasekar, *A Study on Image Segmentation Method for Image Processing*, 12 2021.

[16] A. Jain, *Fundamentals of Digital Image Processing*, ser. Prentice-Hall information and system sciences series. Prentice Hall.

[17] J. L. Mundy and A. Zisserman, "Appendix - projective geometry for machine vision," 1992.

[18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[19] A. Heyden and K. Åström, "Euclidean reconstruction from image sequences with varying and unknown focal length and principal point." 01 1997, pp. 438–443.

[20] D. Brown, "Close-range camera calibration," *Photogramm. Eng.*, vol. 37, 12 2002.

[21] A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," vol. 1, 02 2001, pp. I–125.

[22] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[23] N. I. Chernov and C. Lesort, "Least squares fitting of circles," *Journal of Mathematical Imaging and Vision*, vol. 23, pp. 239–252, 2005.

[24] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, pp. 1115–1138, 12 1991.

[25] O. Semeniuta, "Analysis of camera calibration with respect to measurement accuracy," *Procedia CIRP*, vol. 41, pp. 765–770, 12 2016.