Invariant-Based World Models for Robust Robotic Systems Demonstrated on an Autonomous Football Table

Jordy Senden, Kevin Jebbink, Herman Bruyninckx, and René van de Molengraft

***Abstract*—This work explains the use of invariants in robotic perception and control skills. An "invariant" is a mathematical constraint that remains unchanged under a particular transformation in a system. This property makes robotic functionalities more robust against "disturbances" that cause these transformation. These invariants are stored in a world model (WM), which has a central role in the information architecture of the robot to share information between components. A "robotic" football table serves as an example to illustrate the effectiveness of invariants. Despite looking different, all standard football tables satisfy the same set of invariants; the common layout of the field, line markings and puppets, which are not identical but satisfy the same set of constraints, such as parallelism, partial ordering, and relative colour and intensity differences between objects. During initialization, the invariants are actively identified and saved to the world model. During game play this world model is used by the perception and action skills and is updated when necessary. This work shows how the use of invariants creates robustness against variation in placement of the perception system and against variation in table dimensions and colours. When the camera is misaligned or moved mid-play, the world model is updated to ensure a smooth continuation of the game. The approach is tested on three standard football tables, with different dimensions and colours, showing that the approach is robust on standard tables that adhere to the invariants.**

***Index Terms*—Calibration and identification, computer vision for automation, mapping, object detection, segmentation and categorization, semantic scene understanding.**

I. Introduction

ROBOTIC perception- and control skills often rely on an exact parametrization of the problem at hand, where kinematics and dynamics are defined in Euclidean space. In these *geometric maps*, all object poses are expressed in six degrees of freedom (DOFs) in standard units of measurement and defined with respect to some origin. For robots in a controlled environment with limited variations, e.g. an assembly robot, this description can be used for optimization of a certain task, using classical geometry and mathematics. However, relying on this exact parametrization will cause problems for robots dealing with unpredictable and uncontrollable situations. In semi-structured environments, where exact parameters cannot be guaranteed, this approach is restrictive and hinders robustness. Moreover, object detection algorithms that are based on exact colour or size, are also prone to variations in object appearance and surrounding light conditions. Classical computer vision approaches often try to identify the source of these variations and overcome them, e.g. illumination detection [6] and shadow removal [24]. Detection algorithms based on neural network approaches often try to overcome variations by including measurements where the variations occur into the training data-set. This approach is impractical as addressed by [25]: "in order to deal with the combinatorial complexity of real world images the datasets would have to become exponentially large".

Humans in general do not need to know exact object sizes, colour or distances and do not use fixed reference points to perform their daily tasks. Instead of focusing on exact parameters, we focus more on the layout and relative differences between objects. This work aims to create a central world model as a first class citizen in the software architecture. By describing the world on a properly chosen abstraction level, expected parametric variations can be embedded in the semantics of the world model. To create robotic functionalities that are robust against particular disturbances or transformations in the environment, we should look for constraints that remain unchanged under these transformations, the so-called *invariants*. For example, the cross-ratio of four points on a line is invariant under perspective transformations in computer vision, making it robust against the uncertainty in knowing where the camera is positioned with respect to a scene.

The use of invariants as a means to be more robust against variations is shown on the specific use case of a robotic football table. One side of these systems are automated, such that a single person can play the game of table football against a computer. The rods on one side of the table are connected to motors for translational- and rotational movement. The ball and puppets need to be detected to know where to move the rods and when to kick. In [23] an overhead camera is used for tracking, which was later replaced by a camera looking from underneath a transparent

Manuscript received 24 February 2022; accepted 3 June 2022. Date of publication 23 June 2022; date of current version 12 July 2022. This letter was recommended for publication by Associate Editor T. P. Kucner and Editor M. Vincze upon evaluation of the reviewers' comments. *(Corresponding author: Jordy Senden.)*

Jordy Senden, Kevin Jebbink, and René van de Molengraft are with the Department of Mechanical Engineering, TU Eindhoven, 5612AZ Eindhoven, The Netherlands (e-mail: j.p.f.senden@tue.nl; k.s.jebbink@gmail.com; m.j.g.v.d.molengraft@tue.nl).

Herman Bruyninckx is with the Department of Mechanical Engineering, TU Eindhoven, 5612AZ Eindhoven, The Netherlands, and also with the Department of Mechanical Engineering, KU Leuven, B-3001 Leuven, Belgium, and also with Flanders Make, Leuven, Belgium (e-mail: Herman.Bruyninckx@mech.kuleuven.be).

Digital Object Identifier 10.1109/LRA.2022.3185767

playing field [22]. The overhead camera approach was adapted by [2] and [13]. The commonality between these systems is that the approach is developed specific for one fixed system;

- the exact dimensions of the field have to be known to calibrate the mapping from sensor to world value.
- the world is described in Euclidean space with respect to an arbitrarily chosen fixed reference point.
- object detection is heavily colour dependent. Poor colour calibration leads to positioning errors [12].

This work continues on the development of the *Eindhoven University of Technology Autonomous Football Table* (EUTAFT) in [13]. Since this approach is so parameter-dependent, a calibration is necessary every time a parameter is changed. Playing on a different sized field requires a re-calibration of the sensor mapping. A change in object colour or lightning conditions, requires a new colour calibration. Most humans are easily capable of playing a game of table football, regardless of the exact dimensions of the table or colours of the puppets. The objective is to develop a software stack that can be deployed on every standard football table [1]. To achieve this, the invariant constraints of a standard football table are identified and the necessary perception and control skills are re-developed, based on these invariants.

The methodical approach behind this use case transfers to many other robotic application contexts. Not in the least because invariants fit very well into "world model" components in robotic systems. A world model plays a central role in a system's information architecture, not only to store the "state" of the world, but also to embed functionalities that are shared between several system components: state estimators, monitors, logging, etc. And hence also the semantic information about which invariants are connected to which activities and operations in the world model; like the above-mentioned computer vision functionalities.

First, Section II discusses the symbolic world model of the football table, and how the invariants are used to develop the necessary skills to play the game. Section III explains the experiments that are conducted to show the robustness of this approach, and discusses the results. Section IV elaborates on how this work can be used in a broader context.

II. Method

Human players do not bother with the exact size of the field or colour of the ball or shirts. They also do not care about a fixed reference frame to which they try to position. We can pass a ball to a teammate that is close by, regardless of where exactly this is on the field. To achieve similar flexibility in a robotic system, the variations discussed should be incorporated in the semantics of the world model. Instead of creating skills that are based on exact parameters, they should be based on semantics that do not change between different games, which are referred to as *invariants*. Invariants can be of an absolute form, e.g. the shape of an object, or of a relative form, describing the difference between object properties. For a game of table football, shown schematically in Fig. 1, these invariants are assumed to be:

Absolute invariants:

- There are 2 teams, with 11 players per team.

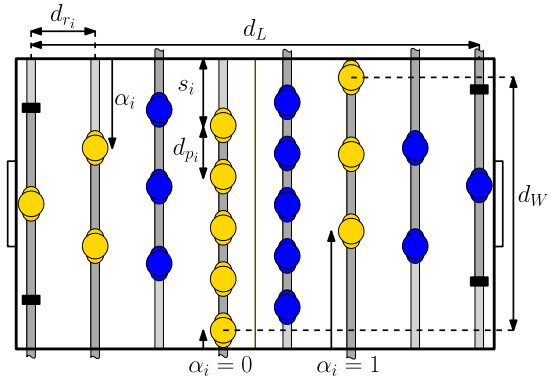

Fig. 1. Schematic overview of the world model of a football table with symbolic measurements.

- The arrangement of the puppets is fixed; 1 keeper, 2 defenders, 5 midfielders and 3 attackers.
- The puppets are connected to rods. Each rod has the freedom to translate along the lateral direction of the field and rotate around its own axes.
- The puppets are not able to translate or rotate with respect to the rods they are connected to.
- The ball is spherical

Relative invariants:

- The field is rectangular, with a length greater than the width.
- The rods are evenly distributed over the length of the field.
- The upper part of the puppet, representing its head, is shorter than the lower part, representing its lower body.
- Puppets on one rod are spaced apart by the same distance.
- The distance between puppets might vary between rods.
- Puppets on one rod have no angular offset to each other.
- The stroke of each rod might vary, but is always larger than the distance between the puppets on that rod.
- The stroke of the 1-rod is at least equal to the goal width.
- The distance between the rods is at least twice the length of the lower part of the puppets.
- The height of the rods from the field is at least the length of the lower part of the puppets and at most this length plus the radius of the ball.
- The colour of the field is uniform, with possible white lines.
- The colour of the puppets is uniform across each team.
- Opposing teams have distinctly different colours.
- All rods have the same colour.
- The ball is of similar size to the puppets.
- The ball has a different colour from the field and players.
- The borders of the field (walls) have a different colour from the field, players, and ball.

The following subsections will give an overview of the EUTAFT and explain how the invariants are used to design the necessary skills to play the game.

A. System Overview

A schematic overview of the EUTAFT is shown in Fig. 2 [13]. The four rods on one side of the table are connected to two motors each, respectively providing the rod's translation and rotation. Each motor is equipped with a quadrature encoder to track its

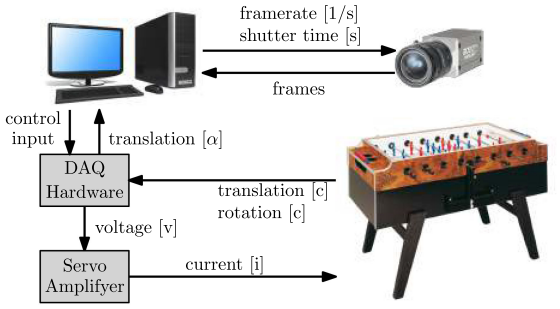

Fig. 2. Overview of connections between the system components and indication of data flow.

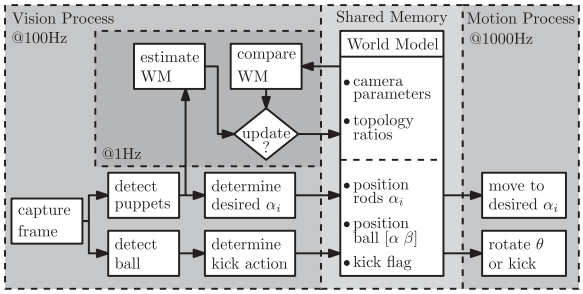

Fig. 3. Overview of software components and data flow, divided into separate processes which run at different rates.

motion. These sensor signals, as well as the outgoing control signals, are processed by the control hardware. An RGB camera is placed above the field to capture images of the game. These are sent to the computer for processing to detect the ball and puppets. Based on the result of this detection, a control output is calculated and sent, via the control hardware, to the amplifiers that power the motors of each rod.

B. Software Overview

To play the game, basic skills need to be implemented such as detecting the playing field, the ball, and the puppets. Next to the detection skills, the puppets must be controlled to block, shoot and pass the ball. An overview of how the skills are implemented in the software architecture is shown in Fig. 3.

Two processes run in parallel during active play; one process detects the objects, while the other controls the actuators. These processes are dependent on each other for information, which is shared through the central world model. The communication between the processes is done through shared memory.

C. Strategy

A strategy is necessary to determine when and where to move each rod. In this work, each of the four rods is translated such that one of the puppets is aligned with the ball in the lateral direction. Since the workspace of neighboring puppets might overlap, the one closest to the ball is chosen. When the ball is located in the reachable space of the puppets in the longitudinal direction, the bar is rotated to kick the ball. A schematic overview of the workspace of the puppets is shown in Fig. 4.

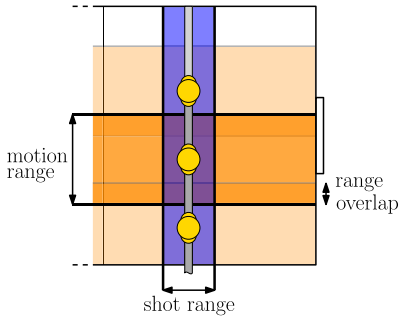

Fig. 4. Schematic indication of workspace of a puppet. The lateral range (indicated in orange)is used for positioning with respect to the ball, the longitudinal range (in blue) is used to determine when to kick the ball.

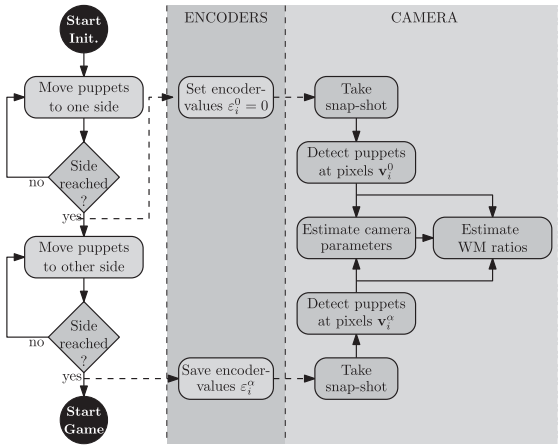

Fig. 5. State- and data flow during the initialization process.

D. Initialization

Instead of measuring exact parameters, the system only has knowledge of the invariants. Before the game starts, initialization is necessary to configure the skills. From the camera standpoint, it is yet unknown which direction 'forward' is and the position and rotation of the puppets at startup are not known. The range of each rod should be determined in both encoder counts and pixel values, since there is no mapping to Euclidean space anymore. The initialization process is visualized in Fig. 5.

First, all four rods are moved to one side of the table, by applying a current to the translation actuators. When the encoders do not register movement anymore it means that the rods are at one end of their motion range. At this point, the encoder values are set to 0 and a frame is captured with the camera. Next, the rods are moved towards the opposite side of the table until the wall is reached. The encoder values at this position are saved and another frame is captured. Fig. 6 shows an overview of these frames. After detecting the puppets, the positions A,B,C, and D are used for camera calibration and estimating the size ratios.

1) Identifying Puppets: A background subtraction method [11] is used on the saved frames to identify which objects are moved in the images. There should at least be eleven distinguishable puppets of the same colour. The value

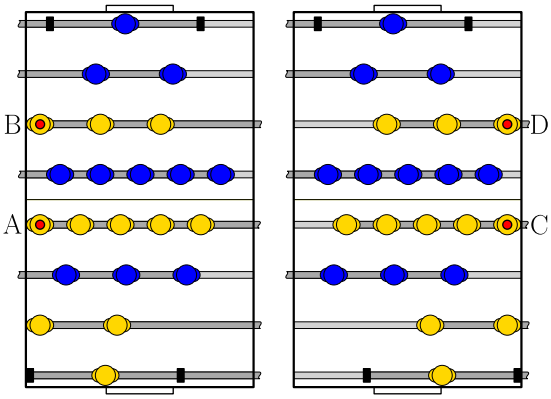

Fig. 6. Visualization of the table's layout during two initialization steps. The positions indicated with A,B,C and D are used for camera calibration and estimating the size ratios.

of their colour (in HSV colour space) is stored in the world model. This colour value is used to group neighboring pixels into blobs [17], which are regarded as potential puppets. By comparing the blobs to the known layout of the puppets, false positives are omitted. Since the pose of the camera is unknown, the projection of this layout to the image is unpredictable. The collinearity property of the puppets is used to find the 5-bar in the image; a set of collinear points in the real world, will appear collinear in the projected image. The cross-ratio (1) of a set of four collinear points will be constant, independent from the camera perspective [16]. Since the five midfielders are equally spaced apart, the cross-ratio of four neighboring midfielders will be $c_r = 4/3$. This is used to determine which five blobs are most likely to correspond to the midfield rod.

$$c_r = \frac{AC \times BD}{BC \times AD} \tag{1}$$

When the puppets of the 5-rod are detected, the puppets of the 3-rod can be found. There should be three blobs of similar colour, which are collinear and represent a line that has a similar angle in the image as the 5-rod. When both the 5-rod and 3-rod puppets are found, finding the most probable blobs that represent the defenders and keeper is straightforward. If all puppets are detected in both frames, the size ratios that make up the scale-invariant world model can be estimated and the camera can be calibrated.

2) Unsupervised Active Camera Calibration: The camera registers a 2D pixel image of the 3D football table. Assuming a pinhole camera [27], this mapping can be modeled by (2).

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R}|\mathbf{T}] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

Here, s is the projective transformation's scale factor, u and v a pixel location in the image and $[x\ y\ z\ 1]^T$ a 3D location in the world. Matrix $\mathbf{R}$ and $\mathbf{T}$ respectively describe the rotation and translation of the camera with respect to the world. The intrinsic camera matrix $\mathbf{K}$, given in (3), contains the geometric properties of the camera.

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Where u_0 and v_0 represent the principal point in the image, which is often assumed to be the center of the image. For modern cameras, the focal lengths f_x and f_y are assumed to be equal ($f_x = f_y = f$) and the axis skew is non-existent ($\gamma = 0$) [10]. Camera lenses introduce image distortion, where radial distortion is the most common [27]. The image needs to be undistorted to get a good estimate of the focal length f and for the pin-hole camera model (2) to hold. This is often done with the checkerboard method, using an even-order polynomial model [3]. However, this work proposes an unsupervised approach that uses the known scene for calibration, similar to [14] and [15], with the assumption that the curves that appear in the undistorted image actually come from straight lines in the world. We use a one order division model, which provides a more accurate approximation while needing less parameters [8]. This model transforms distorted pixel coordinates $\mathbf{v_d} = [u_d\ v_d]^T$ into undistorted coordinates $\mathbf{v_u}$ according to (4).

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

Here, r_b represents a distance between the undistorted pixel and the principal point in the image, calculated as (5), and λ is the distortion parameter.

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

An image of a football table will have an abundance of straight lines, e.g. the lines on the field or the eight rods, which can appear curved in the distorted image. Line segments are extracted from the image using a Canny edge detection [4], followed by a Harris corner detection [9]. The set of curved line segments can be approximated as circular arcs, with center-point $\mathbf{c} = [c_u\ c_v]^T$ and radius c_r. The Taubin circle fit [19] is used as initial guess for a Levenberg Marquardt fit [5], which gives the best fit according to [21]. When circles are fitted for all line segments, their λ's can be estimated with (6).

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

A weighted average is taken over all λ's, where longer segments weigh more heavily, to estimate a single λ for this lens. When λ is found and the images are undistorted, the focal length f can be estimated to complete the intrinsic matrix K. According to [26] the perspective deformation of a known rectangle, as visualized in Fig. 7, can be used for this estimation. The focal length f should adhere to (7), where $\mathbf{v_a} - \mathbf{v_d}$ represents the pixel coordinates of the four corners, with $\mathbf{v_i} = [u_i\ v_i\ 1]^T$, and n_w and n_h can be found with (8) and (9).

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

$$\mathbf{n_w} = \frac{(\mathbf{v_a} \times \mathbf{v_d}) \cdot \mathbf{v_c}}{(\mathbf{v_b} \times \mathbf{v_d}) \cdot \mathbf{v_c}} \cdot \mathbf{v_b} - \mathbf{v_a} \tag{8}$$

$$\mathbf{n_h} = \frac{(\mathbf{v_a} \times \mathbf{v_d}) \cdot \mathbf{v_b}}{(\mathbf{v_c} \times \mathbf{v_d}) \cdot \mathbf{v_b}} \cdot \mathbf{v_c} - \mathbf{v_a} \tag{9}$$

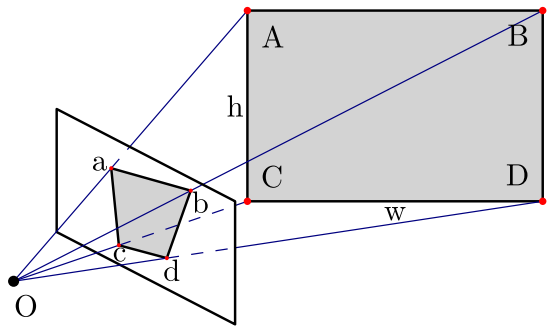

Fig. 7. Deformation of a rectangle in a quadrilateral due to perspective effect.

Writing (8) and (9) into (7) and solving for f, results in 10.

$$f^2 = -\frac{1}{n_{w_3} n_{h_3}} \left[n_{w_1} n_{h_1} + n_{w_2} n_{h_2} - (n_{w_1} n_{h_3} + n_{w_3} n_{h_1}) u_0 - (n_{w_2} n_{h_3} + n_{w_3} n_{h_2}) v_0 + n_{w_3} n_{h_3} {u_0}^2 n_{w_3} n_{h_3} {v_0}^2 \right] \tag{10}$$

Where n_{w_i} and n_{h_i} are the ith component of the vectors in (8) and (9). From (10) it is clear that f cannot be estimated when $n_{w_3} = 0$ or $n_{h_3} = 0$. This happens when the camera is perfectly perpendicular to the field and there is no perspective deformation, i.e. the field will appear as a rectangle in the frame. There is only a rotation θ of the field with respect to the viewing direction of the camera, resulting in a rotation matrix $\mathbf{R}$ as shown in (11). This simplifies (2) to (12), showing f becomes just another scale factor.

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{s}{f} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}^{-1} \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} \tag{12}$$

Estimating the values of the rotation matrix $\mathbf{R}$ is known as a perspective-n-point problem (PnP). 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 [7]. To solve the PnP problem, the SQPnP algorithm as described in [20] is used, which determines the global minimum of the PnP problem at a low computational cost. The knowledge of the size ratios of the table is necessary to be used as reference.

3) Estimating Size Ratios: To estimate the rotation matrix $\mathbf{R}$, the size ratios must be estimated from a possible skewed projection. To overcome this problem, the projective invariant cross-ratio is used together with knowledge of the invariants as discussed in Section II-D2. The cross-ratio between the four detected points of the two frames shown in Fig. 6 is calculated in (13), where $\mathbf{v_a}$, $\mathbf{v_b}$, $\mathbf{v_c}$, and $\mathbf{v_d}$ represent the pixel positions of the four indicated puppets.

$$\frac{||\mathbf{v_a} - \mathbf{v_c}|| \cdot ||\mathbf{v_b} - \mathbf{v_d}||}{||\mathbf{v_b} - \mathbf{v_c}|| \cdot ||\mathbf{v_a} - \mathbf{v_d}||} = c_r \tag{13}$$

The pixel positions of two neighboring puppets, on the same rod i, in both frames should adhere to the same cross-ratio c_r. Knowing that the stroke length should be greater than the distance between two neighboring puppets $s_i > d_{p_i}$, the cross-ratio can be written as (14), where $c_r > 1$ since both s_i and d_{p_i} are positive. Moreover, all puppets except the keeper are able to reach both sides of the table, which results in (15).

$$\frac{{s_i}^2}{(s_i - d_{p_i})(s_i + d_{p_i})} = c_r \tag{14}$$

$$d_W = s_i + (n_i - 1) d_{p_i} \tag{15}$$

Combining (14) and (15) gives expressions for ${d_{p_i}}/{d_W}$ and ${s_i}/{d_W}$, respectively in (16) and (17).

$$\frac{d_{p_i}}{d_W} = \sqrt{\frac{c_r}{c_r - 1}} + (n_i - 1) \tag{16}$$

$$\frac{s_i}{d_W} = \frac{c_r}{c_r - 1} + \sqrt{\frac{c_r}{c_r - 1}} (n_i - 1) \tag{17}$$

The relative table length d_L and rod distance d_r are related through (18). The found points $\mathbf{v_a}$-$\mathbf{v_d}$ in Fig. 6 form a quadrilateral. These are used to calculate the aspect ratio of the rectangle as in (19), similar to the method discussed in Section II-D1.

$$d_L = 7 d_r \tag{18}$$

$$\left(\frac{2 d_r}{d_W} \right)^2 = \frac{\mathbf{n_w}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{n_h}}{\mathbf{n_h}^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{n_w}} \tag{19}$$

After calculating the size ratios, the relative position of each puppet except the keeper is known. The ten known positions of the field player can be used to solve the PnP problem to estimate the rotation matrix R. Once this is known, all parameters of the camera model (2) are identified. The position and stroke of the keeper in the relative world model can be found using the camera model.

4) Identifying Ball: Since the ball is spherical, its projection in the image will be circular. The size of this circle will be similar to the size of the puppets in the image. Moreover, the ball will only be inside the playable region, potential blobs outside this region are omitted. K-means clustering of the image reduces the colour palette of the image [17], making edges more clearly defined. Canny edge detection is performed on the resulting image [4]. From the resulting set of edges, the closed edges with a circular shape and expected size are extracted. Circles that are located outside the field or correspond to known objects, e.g. puppets, are omitted. The remaining circles possibly correspond to the ball. After initial detection, its hue-value is saved in the world model. For the remainder of the game this colour value is used for detection, since the previously described filter steps are too computationally expensive to run at a high frame rate. If the ball is found in the image, the camera model (2) is used to project the ball position into the relative world model.

III. Experiments and Results

To test this approach, experiments are conducted where variations are introduced to the system. One common variation is the

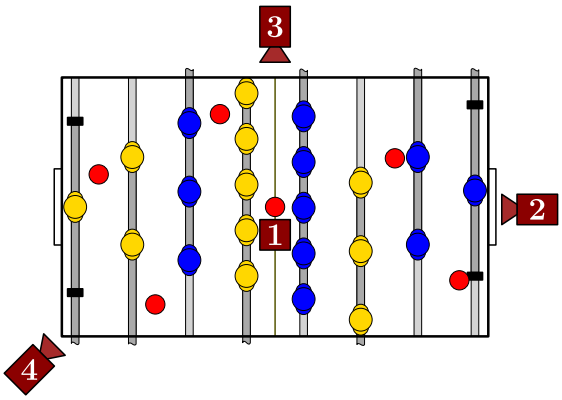

Fig. 8. Schematic overview of the four different camera positions and six ball positions.

TABLE I
ESTIMATED CAMERA PARAMETERS FOR FOUR DIFFERENT POSES

$\mu \pm \sigma$ [mm]	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

camera position with respect to the field. The position can change after detaching or due to vibrations in the system. Secondly, the entire approach is tested on three different tables, with different dimensions and colours.

A. Varying Camera Position

To test the robustness against a varying camera position, four different camera poses are tested, shown in Fig. 8. All experiments are repeated ten times for each pose, resulting in forty measurements in total.

First, the ratio-based estimate of the camera model is validated by comparing it to the camera model found using the standard checkerboard method [27]. Both methods should yield similar values for the intrinsic camera matrix. Secondly, the estimation of the size ratios is validated by comparing them to the actual ratios, which are measured with a tape-measure. Finally, the accuracy of the ball tracking and actuator response is tested. The main goal of the system is to accurately track the ball and formulate a correct response with the puppets. The end result of these algorithms should be able to position a puppet directly in front of the ball, in order to kick it straight ahead. This is possible when the relative positioning accuracy is at least half the width of the puppet's feet. To test this the ball is placed at six positions spread over the field, which are measured by hand. The outcome of the detection algorithm is converted to distances using the size ratios and compared to these measurements.

At each of the four camera position, the focal length f and distortion parameter λ of the camera is estimated as explained in Section II-D2. The results of these estimations are shown in Table I. The focal length is compared to the values found with the checkerboard calibration method, which resulted in $f = 1152$.

For position 1 it is not possible to estimate a focal length f since there is no perspective effect, as discussed in Section II-D2.

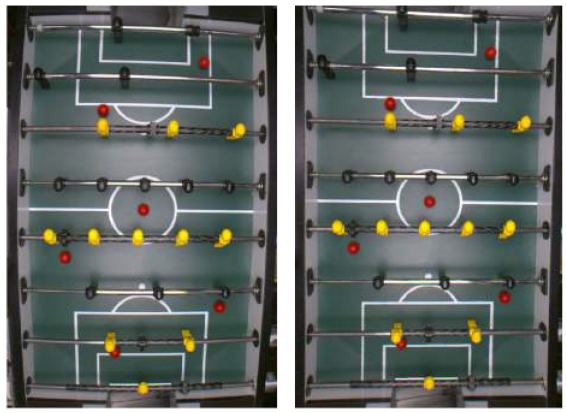

Fig. 9. Left shows the distorted original image. Right shows the result of undistorting with distortion parameter $\lambda = 1.53e-7$.

TABLE II
ESTIMATED SIZE RATIOS OF EUTAFT

$1/d_W[-]$	s_1	s_2	s_3	s_4	d_1	d_2	d_3	d_4
real (10^{-3})	277	648	247	416	362	352	188	292
$\mu(10^{-3})$	281	653	245	419	360	347	189	291
$\sigma(10^{-4})$	5	13	4	7	2	13	1	4

For positions 2 and 4 f deviates less than 25 from the checkerboard method, which is a common offset for this method [18], and are therefore deemed good estimates. Position 3 shows a large offset from the checkerboard value and a high standard deviation between the measurements. This suggests that this pose does not result in enough perspective deformation to calculate an accurate estimate. The model used in the checkerboard method does not contain a parameter λ, making direct comparison impossible. However, Table I shows a consistent estimate for λ throughout the different camera positions. Applying the value $\lambda = 1.53e-7$ to the image shows visually that the distortion is rectified, as shown in Fig. 9.

With these parameters, the camera model is used to estimate the size ratios of the table. These results are shown in Table II and compared to the ratios that are measured by hand for this table. The largest difference in ratio is 0.006 (or 0.38 cm since $d_W = 64$ cm) for this table, which is well within the limit of half the width of the puppet's feet (1 cm).

The relative positions of the six balls are estimated from all four camera poses and compared to the exact position that was measured by hand, shown in Fig. 10. Camera 3 shows the largest error in the position estimation of the ball, this is attributed to the bad focal length estimation.

After the relative world model is estimated and the ball is found, the software suggests a relative position α_i for each rod. The positioning accuracy is tested by using these α's and calculating what the alignment error between the puppets and the ball would be, based on exact measurements of the table. The results of the EUTAFT are shown in Fig. 11. The figure shows that in most cases the ball would be hit straight on and in almost all cases the ball would be touched, even for camera pose 3.

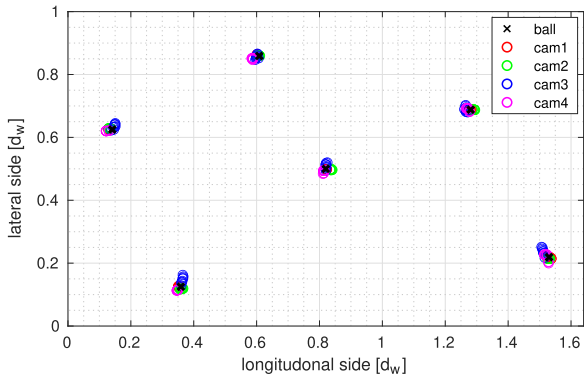

Fig. 10. Results showing the estimated ball positions for 4 different camera setups and 6 different ball positions.

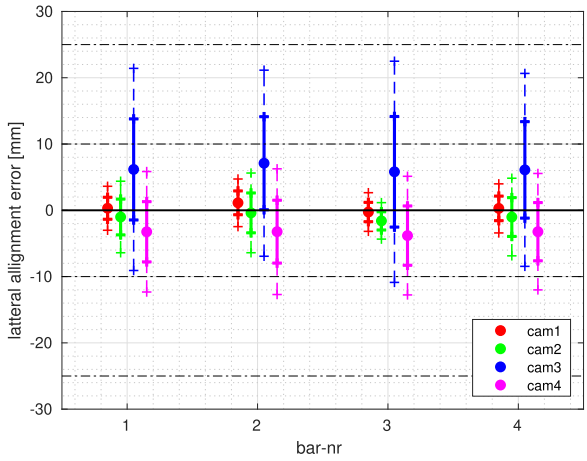

Fig. 11. Alignment error (ϵ_i) of the reference position of each rod with respect to the ball. The average error as well as one- and two times the standard deviation is shown for all four camera positions. The horizontal black lines indicate the ranges in which the ball would be hit straight on ($-10 \leq \epsilon_i \leq 10$) or at least touched ($-25 \leq \epsilon_i \leq 25$).

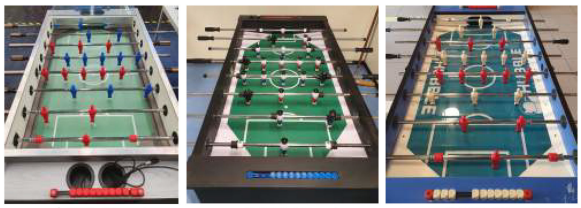

Fig. 12. Pictures of the three different test fields.

TABLE III
AVERAGE ALIGNMENT ERROR OF ALL RODS WITH RESPECT TO THE BALL, SHOWN FOR ALL FOUR CAMERA POSITIONS ON EACH TABLE. A FIXED FOCAL LENGTH WAS USED OF $f = 1152$

$\mu \pm \sigma$[mm]	Cam 1	Cam 2	Cam 3	Cam 4
EUTAFT	0.4 ± 1.8	-1.0 ± 2.8	-9.3 ± 3.3	-3.3 ± 5.6
Table A	-4.3 ± 6.0	-2.4 ± 6.4	-2.7 ± 2.6	3.0 ± 3.0
Table B	-4.1 ± 4.4	-0.7 ± 4.2	-1.7 ± 4.8	3.9 ± 2.9
Table C	0.5 ± 2.5	-0.2 ± 2.5	0.1 ± 3.0	1.2 ± 4.3

Fig. 13. QR-code with link to video (https://youtu.be/RZkiDIedSTA).

B. *Different Football Tables*

This approach should be deployable on every standard football table that adheres to the invariants discussed in Section II. To test this, the previous experiments are repeated on three different football tables, shown in Fig. 12. The aim of these experiments is not to test the estimation of camera parameters, but to get a position reference for the bars that would align the puppets with the ball. Therefore, a fixed focal length of $f = 1152$ is used, instead of estimating one.

Since these football tables are not automated, a human stands in to control the rods after receiving instructions from the computer. Table III shows the desired rod positions, calculated to centimeters, as suggested from the four camera position. This shows that the approach is able to position the puppets accurate enough to diverge or stop the ball.

The results indicate that the approach is capable of localizing the ball and positioning the puppets within the error limits to be able to kick or touch the ball. The best proof to show that this approach is robust against variations is to play a game against the computer. Independent of where the camera starts, the system will automatically initialize its world model and sensors. During active game play the camera is moved without disturbing the game. This is shown in the video which can be viewed by following the link or scanning the QR-code in Fig. 13.

IV. CONCLUSION

This work shows a way of embedding variations into the semantics of the world model, applied to the use-case of a football table. This approach shows robustness against variations like dimensions and colour. The focus of this research was on the object detection- and world modeling part of the problem. The motion-planning and low-level motion-control of the puppets are not discussed. Next to the self-calibration of the sensors, like the camera and encoders, the motion should also be calibrated. Calibration of the distortion of the camera in this work relies on the assumption that the curves found in the raw images should be straight lines. This assumption does not always hold, e.g. if there is a center circle drawn on the field. A way to deal with this is to actively move the puppets and track them with the camera. The consecutive positions found should be in a straight line, since the rods move linearly, in accordance with the invariants. A

future challenge of stepping away from geometric world models, described in Euclidean space, is to create adaptive dynamical models. Since there is no knowledge of physical parameters like distance (and consequentially velocity and acceleration), standard well-known laws of physics that are based on these parameters cannot be used anymore. This approach of creating a world model on a higher level of abstraction than purely geometric is expected to be useful in many application domains where task performance is not measured by geometric errors.

REFERENCES

[1] T. Yore, ITSF Rule Book, Accessed: May 9, 2022. [Online]. Available: https://www.tablesoccer.org/rules/documents/2016_Rulebook.pdf

[2] M. Aeberhard, S. Connelly, E. Tarr, and N. Walker, "Single player foosball table with an autonomous opponent," Georgia Inst. Tech., School Elect. Comput. Eng., 2007.

[3] D. Brown, "Close-range camera calibration," *Physics*, 1971.

[4] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[5] N. I. Chernov and C. Lesort, "Least squares fitting of circles," *J. Math. Imag. Vis.*, vol. 23, pp. 239–252, 2005.

[6] G. Finlayson, C. Fredembach, and M. Drew, "Detecting illumination in images," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, vol. 1, pp. 1–8.

[7] A. M. 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, pp. 381–395, Jun. 1981.

[8] A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, CVPR, 2001, vol. 1, pp. I–125.

[9] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vis. Conf.*, 1988, pp. 147–151.

[10] A. Heyden and K. Åström, "Euclidean reconstruction from image sequences with varying and unknown focal length and principal point," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 1997, pp. 438–443.

[11] A. K. Jain, *Fundamentals of Digital Image Processing* (Prentice-Hall Information and System Sciences Series). Englewood Cliffs, NJ, USA: Prentice Hall, 1988.

[12] R. Janssen, M. Verrijt, J. de Best, and R. de Molengraft, "Ball localization and tracking in a highly dynamic table soccer environment," *Mechatronics*, vol. 22, no. 4, pp. 503–514, 2012.

[13] R. Janssen, J. de Best, R. van de Molengraft, and M. Steinbuch, "The design of a semi-automated football table," in *Proc. IEEE Int. Conf. Control Appl.*, 2010, pp. 89–94.

[14] R. Melo, M. Antunes, J. P. Barreto, G. Falcao, and N. Goncalves, "Unsupervised intrinsic calibration from a single frame using a 'plumb-line' approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 537–544.

[15] P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 3685–3691.

[16] J. L. Mundy and A. Zisserman, "Appendix - projective geometry for machine vision," *Geometric Invariance in Computer Vision*, J. L. Mundy, A. Zisserman, D. G. Bobrow, M. Brady, R. Davis, and P. H. Winston, Eds., Cambridge, Massachusetts: MIT Press, 1992, pp. 463–519.

[17] S. Prabu and J. M. Gnanasekar, "A Study on image segmentation method for image processing," *Recent Trends in Intensive Computing*, M. Rajesh, K. Vengatesan, M. Gnanasekar, R. Sitharthan, A. B. Pawar, P. N. Kalvadekar, and P. Saiprasad, Eds., IOS Press, Dec. 2021, pp. 419–424, doi: 10.3233/APC210223.

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

[19] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Dec. 1991.

[20] G. Terzakis and I. A. Manolis Lourakis, *A Consistently Fast and Globally Optimal Solution to the Perspective-N-Point Problem*. Berlin, Germany: Springer, 2020.

[21] A. Wang, T. Qiu, and L.-T Shao, "A simple method of radial distortion correction with centre of distortion estimation," *J. Math. Imag. Vis.*, vol. 35, pp. 165–172, Nov. 2009.

[22] T. Weigel, "KiRo - a table soccer robot ready for the market," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 4266–4271.

[23] T. Weigel and B. Nebel, "KiRo - an autonomous table soccer player," in *RoboCup 2002: Robot Soccer World Cup VI*. G. A Kaminka, P. U. Lima, and R. Rojas, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 384–392, doi: 10.1007/978-3-540-45135-8_34.

[24] J. Xiang, H. Fan, H. Liao, J. Xu, W. Sun, and S. Yu, "Moving object detection and shadow removing under changing illumination condition," *Math. Problems Eng.*, vol. 2014, pp. 1–10, 2014.

[25] A. L. Yuille and C. Liu, "Deep Nets: What have they ever done for vision?," *Int. J. Comput. Vis.*, vol. 129, no. 3, pp. 781–802, 2021.

[26] L.-W. He and Z. Zhang, "Whiteboard scanning and image enhancement," *Digit. Signal Process.*, vol. 17, no. 2, pp. 414–432, Mar. 2007, doi: 10.1016/j.dsp.2006.05.006.

[27] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.