Robot at Factory 4.0: An Auto-Referee Proposal Based on Artificial Vision

Tony Ferreira[1(✉)], João Braun[1,2,3,4], José Lima[2,3,4], Vítor H. Pinto[1,5], Murillo Santos[6], and Paulo Costa[1,4]

[1] Faculty of Engineering of University of Porto, Porto, Portugal
{up201706545,vitorpinto,paco}@fe.up.pt, jbneto@ipb.pt
[2] Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, Portugal
jllima@ipb.pt
[3] Laboratório para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança, Bragança, Portugal
[4] INESC TEC, Centre for Robotics in Industry and Intelligent Systems, Porto, Portugal
[5] SYSYTEC (DIGI2) - Research Center for Systems and Technologies (Digital and Intelligent Industry Lab), Porto, Portugal
[6] CEFET-MG, Campus Leopoldina Departamento de Eletroeletrônica, Leopoldina, Brazil
murillo.ferreira@cefetmg.br

Abstract. The robotization and automation of tasks are relevant processes and of great relevance to be considered nowadays. This work aims to turn the manual action of assigning the score for the robotic competition Robot at Factory 4.0 by an automatic referee. Specifically, the aim is to represent the real space in a set of computational information using computer vision, localization and mapping techniques. One of the crucial processes to achieve this goal involved the adaptive calibration of the parameters of a digital camera through visual references and tracking of objects, which resulted in a fully functional, robust and dynamic system that is capable of mapping the competition's objects accurately and correctly performing the referee's tasks.

Keywords: Engineering education · Robot at factory · Camera calibration · Pose estimation · Computer vision · STEM · Education 4.0

1 Introduction

Over time, technology's continuous growth and adaptation are encountered, whose usefulness falls into a broad and diverse horizon of applications. One aspect of this evolution is featured in automated and robotic systems, where their development and implementation grant the ability to cooperate and perform tasks that exceed human limits. This application can be seen in many areas, for example, for industrial purposes, such as industrial manipulators and

D. Tardioli et al. (Eds.): ROBOT 2022, LNNS 589, pp. 475–487, 2023.
https://doi.org/10.1007/978-3-031-21065-5_39

autonomous mobile robots (AMRs), Atlas humanoid and quadrupedal robots developed by Boston Dynamics, or even the robotic vacuum cleaners that you may have at home.

While the areas of automation and robotics are very appealing, their complexity can have an inherently intimidating aspect from the viewpoint of robotics enthusiast spectators. Thus, competitions can be held to provide reasons and interest to participate in a given imposed challenge, allowing the development of knowledge about the subject, such as programming, kinematics, dynamics, control theory, and computer vision. Additionally, robotics competitions develop other skills, such as soft-/hard-/transferable-skills, technological concepts and education through Science, Technology, Engineering and Mathematics (STEM) methodology.

The realization of projects involving robotics and automation concepts, such as the work referred to in this document, is seen as a carrier for representing engineering fields. That is, they generate and captivate interest among technology enthusiasts causing attraction to young audiences. To date, Unmanned Aerial Vehicles are showcased at the International Aerial Robotics Competitions [1] and Mohamed Bin Zayed International Robotics Challenge (MBZIBC) [2]; quadrupedal [3] robots for Autonomous Robots for Gas and Oil Sites (ARGOS); humanoid robots [4] as seen at RoboCup; and marine robots [5] presented in Student Autonomous Underwater Challenge - Europe (SAUC-E), Center for Marine Research and Experimentation (CMRE), and European Robotics League (ERL).

Our focus addresses *augmented reality* [6,7] fiducial markers such as ARTag [8], AprilTag [9], CALTag [10], Circular Data Matrix [11], TopoTag [12], CanTag [13], ArUco [14], STag [15], among others. These are distinguished by attributes and encodings and their application can be *augmented reality* application, such as simultaneous localization and mapping (SLAM) [16,17], reference for the measurement [18], or object tracking [19].

Therefore, this document is within the scope of a Robotics Competition created by members of the INESC TEC [20]. This competition is characterized by a set of participants that try to maximize the score obtained by moving and positioning parts from their original location to the requested ones. Competitors are responsible for moving the parts, with AMRs, from an initial location to one or several desired positions. The previous versions of this competition had a referee responsible for following the competition's rules to perform the scoring task. The goal is to contribute and enhance this technological environment by converting this manual task to an automated one using computer vision methods with an RGB camera and fiducial markers.

2 Auto Referee Development

This chapter will discuss the methodologies used to accomplish the desired functionalities of the system, such as markers, calibration methods and parts detection.

2.1 Description of the RaF 4.0 Competition

The Robot at Factory (RaF) is an academic and international competition within the scope of robotics that is carried out through 3 rounds, with incremental difficulty, of 10 min, where the participants must maximize their score in due time with the robot. One point is incremented each time the part is correctly placed in machinery or output, and each round adds a new Part type, meaning that more destinations, operations, and points are at stake. Finally, if the piece arrives at the final destination, "Outgoing Warehouse", the score is increased according to the obtained points.

It is designed to simulate an on-scale warehouse and factory system where some initially stored components, with distinct types, are moved either to an operating section, "Machine-A/-B", that modifies their properties or the expedition section, which expects the finished and desired product. Throughout this document, the components that are transported will be called "parts", or "pieces", and they can have one of three types: Raw (red), Intermediate (green) and Final (blue), meaning that a part needs two, one or no operations and provides 1, 2 or 3 points, respectively. If no operations are needed, then the piece is placed at the output, and for each correct operation, the piece is updated to the next type, e.g., a raw type piece is updated to intermediate and an intermediate to final the final type.

2.2 Methodology

The developed system that will be responsible for the scorekeeping consists of 5 main stages:

1. **Field Mapping and Intrinsic Calibration** – Identification of the zones corresponding to the ingoing warehouse, machine A, machine B and outgoing warehouse and application of the calibration method to estimate the intrinsic parameters of the Raspberry Pi camera module V2;
2. **Detection of Fiducial Markers** – Use of the detection library for the fiducial marker identification;
3. **Extrinsic Calibration and Position Estimation** – Estimate the position of the pieces on the field according to the camera position in space;
4. **Position Verification** – According to the field mapping and estimated position, verify that it is in the right zone with the correct piece type to assign the score.

In sum, this project aims to track each $9 \times 6 \times 6.5$ cm squared prism pieces and update its pose data to perform the referee scoring task correctly. For this purpose, an ArUco marker is fixed on top of each transported piece granting us the ability to receive helpful information about these objects, such as position, orientation, identifier, and type. This information is decoded by the previously calibrated camera and later sent to the referee system via ZeroMQ (ZMQ). Camera procedures are done with *OpenCV* python libraries on a Raspberry Pi 4 8GB with Thorny IDE as editor, having a raspberry camera module V2 as an

input device. On the other hand, the referee and clock software were done via Lazarus IDE in Free Pascal.

Field Mapping and Intrinsic Calibration

The fundamental goal of the automatic referee is to detect that a particular type of piece is positioned at another specific location. These locations are the input boxes of the machinery-A/-B sections (far left par of rectangles from the middle left and middle right sections) and the outgoing warehouse (bottom right), as showcased in Fig. 1. Note that the measurements are done according to the field referential whose centre is shown with $(0, 0)$. Furthermore, the camera and the pieces' positions are mapped according to the field referential. Following the tabs "Referee" and "Clock" in Fig. 3b, note that this is the starting step of the referee system alongside the pieces starting data that is received and generated from the Clock interface, which is then sent via UDP.

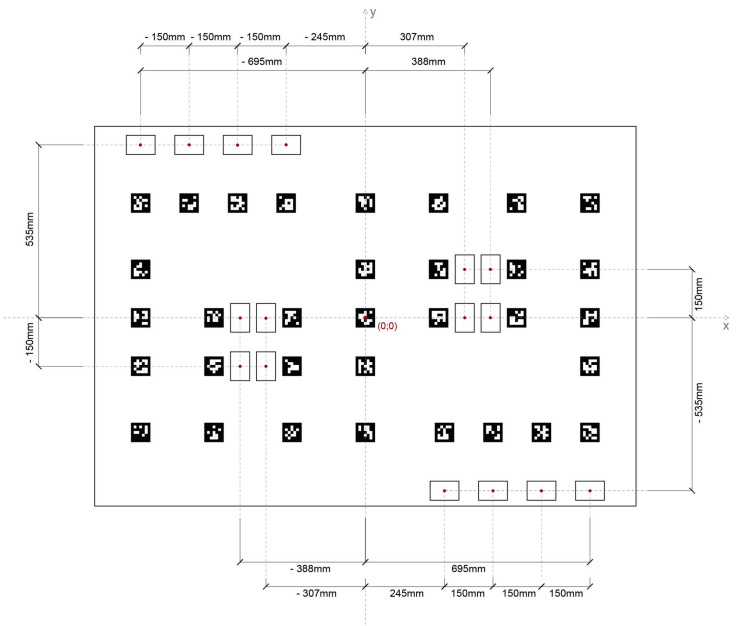


Fig. 1. Robot at factory 4.0 field layout with the sections mapped.

Before the marker detection and pose estimation procedure, the camera must be calibrated, defining the first step of the Raspberry Pi device, as shown on the "RPI" left tab in Fig. 3b. For this purpose, a mathematical representation of the device will be used to have an association between the 3-D world coordinates and the 2-D image plane coordinates. The model is called the *pinhole* camera model and it is described by intrinsic calibration matrix ($\mathbf{K}$) that relates

the lens and its "inside" properties according to the focal length (f), projected image height/width (m), the principal point (u_0, v_0) and the skew coefficient (γ, typically equal to 0); and extrinsic ($\mathbf{R}$ and $\mathbf{T}$) parameters which describes the position of the camera in the world by a rotation and translation matrix.

$$\boldsymbol{K}_{3x3} = \begin{bmatrix} f_x.m_x & \gamma & u_0 \\ 0 & f_y.m_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$[\boldsymbol{T}_{3x1}]^T = [t_x, t_y, t_z]^T \tag{2}$$

$$\boldsymbol{R}_{3x3} = \begin{bmatrix} r_{11}\ r_{12}\ r_{13} \\ r_{21}\ r_{22}\ r_{23} \\ r_{31}\ r_{32}\ r_{33} \end{bmatrix} \tag{3}$$

For this purpose, direct linear transformation [21], autonomous [22] or active [23] are means for calibration. However, the traditional method to estimate the intrinsic parameters whose procedure is supported by visual reference elements was used. Therefore, *Zhang's* "chessboard" method [24] is used with *OpenCV* libraries. Finally, *cv2::calibrateCamera* is used with the pictures taken of the chessboard marker in various orientations, with the Raspberry Pi camera module v2, as shown in Fig. 2.

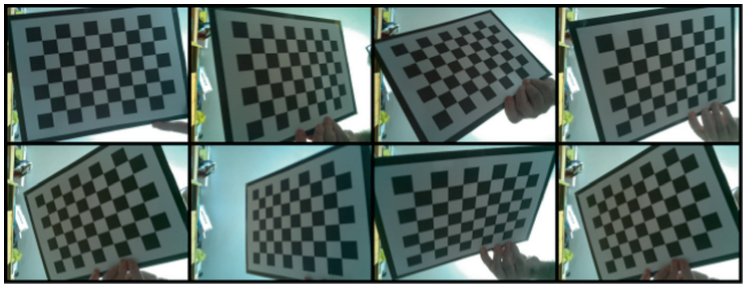


Fig. 2. Pictures of the chessboard marker taken with the Raspberry Pi Camera Module v2 to perform the intrinsic calibration.

Fiducial Marker Detection

In order to obtain the pieces' information, ArUco markers with distinct IDs are fixed on top of each piece. Currently, IDs ranging in the [0, 26], [30, 35] and [50, 65] intervals are already in use on the competition floor. Thus the IDs in the interval [40, 47] are dedicatedly used for piece detection. All the markers used are 5×5 ArUco Tags generated in[1] and the ArUco detection procedure is done with *cv2::aruco.detectMarkers*.

[1] https://chev.me/arucogen/.

However, the camera detects all the ArUco markers within its frame. Since the field markers are used for calibration purposes and the pieces markers are points to be mapped from the 2-D image plane to the 3-D field coordinates, a distinction must be made to assign the data to its correct procedure, resulting in a dataset of "2D & 3D Field Mapping" and "2D Pieces Mapping", as shown at the "RPI" tab in Fig. 3b). The first set is later used for the extrinsic calibration, and the second set is used for the pose estimation procedure according to the intrinsic and extrinsic calibration.

Extrinsic Calibration and Pose Estimation

Hereafter, the extrinsic parameters are ready to be estimated. Since the camera is positioned in an environment characterized by numerous markers, we take advantage of this property by using all the field ArUcos detected markers as input for the extrinsic calibration, with *OpenCV Perspective-n-Point* (PnP) algorithm.

From this point, all the input data needed to perform pose estimation has been acquired, meaning that this procedure is ready to be used if, and only if, $\mathbf{m}$ is greater than $\mathbf{n}$, being $\mathbf{n}$ the minimum number of points needed to compute the position assessment and $\mathbf{m}$ is the number of field points detected within one frame. In order to calculate its position, an ideal *pinhole* camera model will be used with the following equations, according to the camera parametrization:

$$Z \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \boldsymbol{K}_{3x3} \left[\boldsymbol{R}_{3x3} \, \boldsymbol{T}_{3x1} \right]_{3x4} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix} \tag{4}$$

$$\boldsymbol{R}^{-1} \boldsymbol{K}^{-1} Z \begin{bmatrix} \mu \\ \nu \\ 1 \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + \boldsymbol{R}^{-1} \boldsymbol{T} \tag{5}$$

From Eq. 5, we have a model that translates an image point (μ, ν) into world coordinates (x_p, y_p, z_p) from all the parameters gathered from R and T, as illustrated on the previous section. However, one last parameter to be determined is the constant value of the focal plane, Z. Since the piece's height, z_p, is constant and known (65 mm), the last row of the equation is fully defined, resulting Eq. 7. Once determined, the system can translate a point from the camera image plane into its world coordinates. Therefore, if a piece is detected within the current camera frame, then its world projection is estimated according to the field referential and sent to the referee interface via ZeroMQ (⌀MQ), concluding the cycle of the camera-related procedures of "RPI" tab and proceeding the "Referee" procedures, as illustrated in the Fig. 3b.

$$\boldsymbol{R}_3^{-1}\boldsymbol{K}_3^{-1}Z = z_p + \boldsymbol{R}_3^{-1}\boldsymbol{T}_3 \tag{6}$$

$$Z = \frac{z_p + \boldsymbol{R}_3^{-1}\boldsymbol{T}_3}{\boldsymbol{R}_3^{-1}\boldsymbol{K}_3^{-1}} \tag{7}$$

Pose Verification

Hereafter, the pieces' positions are stored and updated according to the current and previous location, with the intent to have a comparison measurement that provides the system with an error that translates the difference between the current and last position with respect to x, ϵ_x and y, ϵ_y. Furthermore, ϵ is obtained as an average value of ϵ_{prev} and ϵ_{curr}, according to the previous and current iterations.

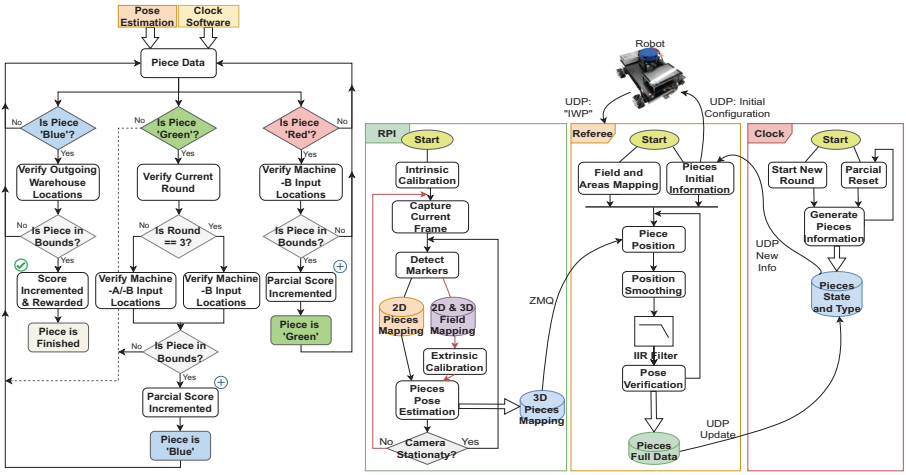
With the intent to reduce the error, a position smoothing procedure was implemented to identify three-piece cases: stationary, moving, and invalid. Thus, a first-order Infinite Impulse Response (IIR) filter with $\alpha \in [0, 1]$ is applied, which grants a fluid motion when accelerating:

$$new_{pos} = curr_{pos}\alpha + prev_{pos}(1 - \alpha) \tag{8}$$

Furthermore, one can note that by analyzing Eq. 8, a higher α benefits the current position, meaning that further displacements between the current and previous positions will result in a least noticeable smoothing. On the other hand, a lower α showcases higher smoothing due to the importance given to the previous position. Therefore, two rates and maximum errors were defined to provide a distinguishing condition for all three states. Hence, an absolute error that is: bellow $max_{\epsilon L}$ results in an $\alpha_H = 15\%$, desired for a higher smoothing of stationary pieces; between $max_{\epsilon L}$ and $max_{\epsilon H}$ provides an $\alpha_L = 70\%$, for lesser smoothing of moving pieces; and greater than $max_{\epsilon L}$ means that is it an unusable and invalid estimation.

Finally, the piece pose verification is done on each detected piece on the field according to if its centre position is within the bounds of a rectangle. If so, it is a correct outcome. Also, if validated, the piece is updated, and a point is rewarded. With this being said, the pose verification is presented in Fig. 3a whose starting point is based on the data gathered from the pose estimation, from the camera, and the pieces initial data, from the clock (end-factor datasets from the "RPI" and "Clock" tab in the Fig. 3b). Hereafter, the piece is distinguished according to its type/colour (raw/red, intermediate/green or final/blue) and round. Note that only the third round is checked since, at this time, the order of machinery is essential if operations are needed (raw and intermediate types). Then, this is followed by verification of the pieces' position and the mapped areas measured at the "Field Mapping" procedure. Finally, if the position is validated, the piece is updated to the succeeding type, and the verification algorithm is redone until the piece is finished.

Additionally, the pose verification procedure concludes the referee cycle from the "Referee" tab, whose results fall on the full and updated pieces information. This data is also updated towards the auxiliary Clock interface via UDP, which provides a grid with all the pieces data. The Referee-Clock interaction is also illustrated in Fig. 3b where the data interchange relation is noticeable.



(a) Flowchart of the referee pose verification algorithm.

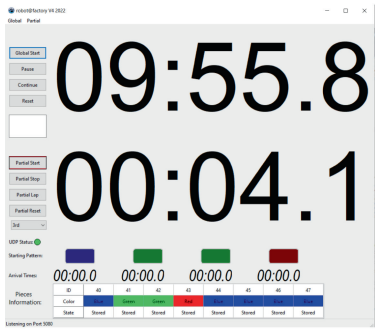
(b) Flowchart of the global architecture interactions.

Fig. 3. Referee verification algorithm and the global system architecture diagram. (a) Flowchart of the referee pose verification algorithm. (b) Flowchart of the global architecture interactions.

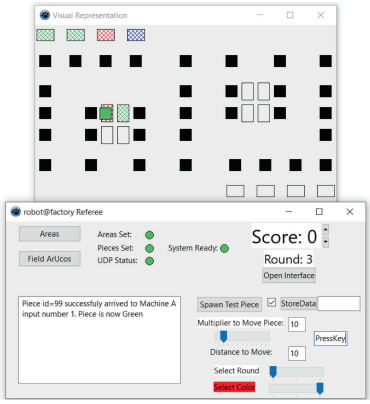
2.3 Interface

The Clock interface is responsible for showcasing the overall state of a given round by presenting the time remaining, the states of the pieces and their timestamps, above the grid, of the pieces' arrival at the "Outgoing Warehouse". As seen in Fig. 4a, the pieces' information is displayed on a grid providing each piece's colour and state according to its ID. The colours denote the pieces type, which the possibilities can be "Blue", "Green", "Red", and "White". The white type was defined to identify correctly delivered pieces at the output. Additionally, the pieces can have one of four states at a given time: "Unassigned", "OnField", "OnProcess" and "Done", meaning that a piece has been reset, detected, on a machine or correctly placed at the "Outgoing Warehouse", respectively.

On the other hand, the referee interface illustrated in Fig. 4b has two windows: the main window, "robot@factory Referee", and the on-time field representation, "Visual Representation". The first window showcases the current score and round of an attempt alongside simulation features at the bottom. This window also has a manual score incrementer and decrementer to avoid erratic



(a) Auxiliary Clock interface.



(b) Referee interface.

Fig. 4. Interfaces of the automatic referee system. (a) Auxiliary clock interface. (b) Referee interface.

actions, for example, wrong pose estimation of a piece resulting in an unnoticed rewarding scenario or rewarding a miss-placed piece.

3 Results

The results section will address the preliminary simulation tests, the pose estimation experiments to stress the localization errors and, finally, will present the validation with an actual mobile robot in the competition environment.

3.1 Pose Estimation Test

The piece pose detection was tested on the actual competition scenario to verify the position estimation procedure from the most challenging position, which is a diagonal point of view (furthest x and y coordinates of the field referential) with the camera positioned at $\smile (-850; 600; 900)_{mm}$ and pointed towards the field centre. Furthermore, the impact of the position smoothing procedure should also be reviewed under the same testing scenario. The pose estimation procedure was done by pre-defining z equal to 65 mm with the pieces located at the machines and outgoing wawrehouse since these are the most relevant sections for the referee pose verification procedure, and thus placed as follows $(x; y; z)_{mm}$:
piece ID 40 $\smile (-388; -150; 65)$; piece ID 41 $\smile (307; 150; 65)$; piece ID 42 $\smile (245; -535; 65)$; piece ID 43 $\smile (695; -535; 65)$.

This test consisted of a continuous detection and pose estimation of the four pieces on-field throughout 1 min, and the results are shown in Fig. 5.

From the "Simple" test case, the influence of the camera-piece relative position is noticeable, which increased the estimation error as the distance also

Simple

ID	Mean Abs(e_x)	Mean Abs(e_y)	Var. e_x	Var. e_y	Detection Rate
40	4,148	3,951	0,280	0,073	89,53%
41	13,025	5,979	1,122	0,336	100,00%
42	1,992	18,531	6,291	2,588	60,07%
43	18,426	18,862	22,717	7,019	52,21%
Total	**9,130**	**10,104**	**7,603**	**2,504**	-

Smoothing

ID	Mean Abs(e_x)	Mean Abs(e_y)	Var. e_x	Var. e_y	Detection Rate
40	1,342	8,802	0,151	0,070	77,09%
41	3,030	4,554	0,030	0,097	71,69%
42	7,994	6,674	0,648	1,030	81,67%
43	9,934	5,549	3,392	1,487	55,97%
Total	**5,340**	**6,496**	**1,055**	**0,671**	-

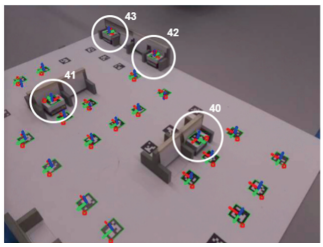


Fig. 5. Position estimation error, variance and number of samples for each detected piece marker from a diagonal perspective without (top) and with (bottom) the position smoothing.

increased. Additionally, by analysis of the variance, the pose estimation procedure provides values with less consistency for farther positioned pieces.

On the other hand, the same test scenario was executed with the IIR filter at the "Smoothing" table. The system significantly improved with this procedure by lowering the maximum error by almost half and enhancing the consistency of all the results due to the decreased variance values. Thus, we conclude that the filter supports the system with additional accuracy, versatility and reliability by filtering and refining the estimated values.

These results are affected by uncontrolled factors such as lighting, shadows, camera resolution, and noise, which were not possible to test in due time. The detection rate showcases are a rough estimation since these are influenced by the uncontrolled factors previously mentioned, the camera frame rate, and the processing power of the device.

3.2 Round 3 Verification

The behaviour and decision-making of the automatic referee will be registered at the most challenging round to validate the developed system. At this attempt, one blue, two greens and one red type piece are displaced initially, and at the end, it should be verified that a total of 8 points will be rewarded (one point for the blue piece, 2 points for each green piece and 3 points for the red piece) if all the correct displacements were recorded.

In Fig. 6, what should be noted are the pieces' position on the visual representation of the referee system and the updated data showcased on the grid of the Clock interface according to the real scenario.

Firstly, at a timestamp of 9:13.0 is displayed a correct red type piece at the Machine-A input, updating its colour to green and its state to "OnProcess". At the bottom right of the field, a correct blue type piece has been correctly placed and thus rewarding 1 point, which updates not only its colour and state to "White" and 'Done" but also the score from the referee interface. Note that one green piece is incorrectly placed at the "Outgoing Warehouse", on the bottom right, on purpose, which did not trigger a state update.

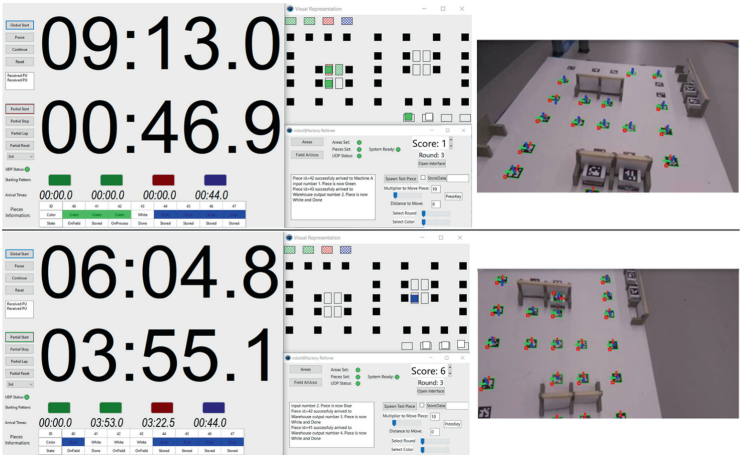


Fig. 6. System validation for piece poses identification in round 3. At each frame, Clock interface at the left and referee system with the camera at the right.

On the next frame, time 6:04.8, one green piece is correctly positioned at the Machine-B input, thus converting the piece ID 40 to its final (blue) type. Beforehand, piece ID 41 and 42 were correctly updated to their final type, and at this current time frame, they are located at the output section and consequently rewarding 5 additional points, 3 for the initially red and 2 for the green type piece, for a total score of 6. The remaining blue piece started as green and contributed with the final 2 points remaining for the maximum score of 8. It is also shown the arrival timestamp of each piece above the grid of the Clock interface.

From the two frames illustrated in Fig. 6, note that the camera is positioned at distinct positions, which showcases the system's ability to dynamically calibrate the camera and estimate the pieces' positions without negatively impacting the system and grating resilience to oscillations and rotations.

4 Conclusions and Future Work

To conclude, a system was developed that can perform the verification and scoring mechanism of the Robot at Factory 4.0 robotic competition. Using Zhang's alongside perspective-n-point methods provided a dynamic calibration with high accuracy for the pose estimation procedure using the field reference for the camera and the pieces' positions. Additionally, improved results were obtained from the position smoothing procedure which granted further versatility and precision. In sum, the automated task of the referee provided successful results whose performance was noticeable. Therefore, the system is adequate to be further improved and become a relevant device for the competition mentioned. However, the system should be further tested under controlled conditions by explicitly

characterizing its impacting variables to improve its capabilities. In the future, additional functionalities can be implemented as trajectory estimation through Kalman Filter, develop statistical data associated with each team and the overall competition, additional pieces and robots' data, or even the implementation of a multi-camera configuration which would grant more accuracy and robustness to the system. Additionally, this system has the potential to be adapted and used in other robotics applications that benefits from localization and mapping purposes under controlled environments.

Acknowledgment. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia, within projects LA/P/0063/2020 and POCI-01-0247-FEDER-072638, co-funded by FEDER through COMPETE 2020. Authors acknowledge 5DPO RobotAtFactory team for making their time and robot available to conduct tests. The project that gave rise to these results received the support of a fellowship from "la Caixa" Foundation (ID 100010434). The fellowship code is LCF/BQ/DI20/11780028.

References

1. Bachrach, A., Prentice, S., He, R., Roy, N.: Range-robust autonomous navigation in GPS-denied environments. J. Field Robot. **28**(5), 644–666 (2011)
2. Spurný, V., et al.: Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. J. Field Robot. **36**(1), 125–148 (2018)
3. Bellicoso, C., et al.: Advances in real-world applications for legged robots. J. Field Robot. **35**(8), 1311–1326 (2018)
4. Jin, X., Yang, Z., Wang, H., Yin, J., Wang, C., Fenghua, W.: Improvements of sycu humanoid robot. J. Phys. Conf. Ser. **1693**, 012204 (2020)
5. Ferri, G., Ferreira, F., Djapic, V. . Fostering marine robotics through competitions: from sauc-e to erl emergency 2018. In: OCEANS 2018 MTS/IEEE Charleston (2018)
6. Shabalina, K., Sagitov, A., Sabirova, L., Li, H., Magid, E.: ARTag, AprilTag and CALTag fiducial systems comparison in a presence of partial rotation: manual and automated approaches. In: Gusikhin, O., Madani, K. (eds.) ICINCO 2017. LNEE, vol. 495, pp. 536–558. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-11292-9_27
7. Zakiev, A., Tsoy, T., Shabalina, K., Magid, E., Saha, S.K.: Virtual experiments on aruco and apriltag systems comparison for fiducial marker rotation resistance under noisy sensory data. In: 2020 International Joint Conference on Neural Networks (IJCNN) (2020)
8. Dos Santos, A.B., Dourado, J.B., Bezerra, A.: Artoolkit and qualcomm vuforia: an analytical collation. In: 2016 XVIII Symposium on Virtual and Augmented Reality (SVR) (2016)
9. Wang, J., Olson, E.: Apriltag 2: efficient and robust fiducial detection. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2016)
10. Atcheson, B., Heide, F., Heidrich, W.: Caltag: high precision fiducial markers for camera calibration. In: VMV 2010 - Vision, Modeling and Visualization (2010)

11. Naimark, L., Foxlin, E.: Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In: Proceedings. International Symposium on Mixed and Augmented Reality (2002)
12. Guoxing, Yu., Yongtao, H., Dai, J.: Topotag: a robust and scalable topological fiducial marker system. IEEE Trans. Visual Comput. Graph. **27**(9), 3769–3780 (2021)
13. Rice, A.C., Beresford, A.R., Harle, R.K.: Cantag: an open source software toolkit for designing and deploying marker-based vision systems. In: 4th Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM 2006) (2006)
14. Li, B., Wu, J., Tan, X., Wang, B.: Aruco marker detection under occlusion using convolutional neural network. In: 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE) (2020)
15. Benligiray, B., Topal, C., Akinlar, C.: Stag: a stable fiducial marker system. Image Vis. Comput. **89**, 158–169 (2019)
16. Rekleitis, I., Meger, D., Dudek, G.: Simultaneous planning, localization, and mapping in a camera sensor network. Robot. Auton. Syst. **54**(11), 921–932 (2006)
17. Darabi, S., Shahri, A.M.: Vision based simultaneous localization and mapping using sigma point kalman filter. In: 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE) (2011)
18. Monkman, G.G., Hyder, K., Kaiser, M.J., Vidal, F.P.: Using machine vision to estimate fish length from images using regional convolutional neural networks. Meth. Ecol. Evol. **10**(12), 2045–2056 (2019)
19. Singh, A.M., Ha, Q.P., Wood, D.K., Bishop, M.: Low-latency vision-based fiducial detection and localization for object tracking. In: Proceedings of the International Symposium on Automation and Robotics in Construction (IAARC) (2017)
20. Costa, P., Lima, J., Pinto, V.: Rules for robotatfactory 4.0 2022. In: Festival Nacional de Robótica Portuguese Robotics Open (2021)
21. Zhao, Z., Ye, D., Zhang, X., Chen, G., Zhang, B.: Improved direct linear transformation for parameter decoupling in camera calibration. Algorithms **9**(2), 31 (2016)
22. Lei, C., Wu, F., Hu, Z., Tsui, H.T.: A new approach to solving kruppa equations for camera self-calibration. In: Object Recognition Supported by User Interaction for Service Robots (2002)
23. Rebello, J., Das, A., Waslander, S.: Autonomous active calibration of a dynamic camera cluster using next-best-view. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
24. Liang, X., Du, Y., Wei, D.: An integrated camera parameters calibration approach for robotic monocular vision guidance. In: 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC) (2019)