scientific reports

Check for updates

OPEN

Calibration of the omnidirectional vision system for robotics sorting system

Ivan Kholodilin, Zihan Zhang✉, Qihui Guo & Maksim Grigorev

Sorting objects is a relevant field in the robotics direction and includes a complex approach in order to meet the objectives. The important part of this system is a vision system. Vision system should provide a correct distance to obstacles to make the robot operate with them properly. In this paper is considered an omnidirectional vision system consisting of the fisheye camera and structured light. Towards other vision systems, an omnidirectional one has its own distinctive advantages as it can provide more information of the environment on the image snapshot. The advantage of the use of the structured light is its simple detection the image. However, before moving to the distance measurements the vision system must be calibrated. This paper proposed an improved extrinsic calibration of the vision system. The calibrated vision system is evaluated experimentally by using simulated and real data to demonstrate that the proposed calibration method is more robust and provides a higher measurement accuracy in comparison to other methods. Obstacle distances were obtained with a measurement error did not exceed 0.5% for synthetic data, did not exceed 1.6% for real data and computation time for the calibration process decreased by 12%. This paper also proposed a simulator which was useful during experiments part. By simulator it was possible to generate datasets of different vision system configurations to test the proposed calibration method and compare it with other methods before moving to real experiments. Built using Unity, the simulator integrates vision system, industrial robot, elements of the environment, and allows one to generate synthetic photo-realistic datasets with automatic ground truth annotations. The proposed simulator and supporting materials are available online: https://github.com/kholodilinivan/RoboticsSystemCalibration.

Keywords Calibration, Measurements, Robotics, Omnidirectional vision, Simulation, Structured light

Sorting objects is an important part of production processes in many fields, such as the food industry[1], electronics manufacturing[2], pharmaceutical industry[3], etc. Using industrial robots for sorting can improve the efficiency and accuracy of the process, as industrial robots can process large number of products in a short time. In general, the use of industrial robots for sorting objects is a relevant direction in the development of industrial automation, which helps to improve the efficiency, quality and safety of enterprises. In the case of sorting objects in a dynamic environment, when objects do not have a fixed position and orientation in world coordinates, the integration of a computer vision system is relevant.

Computer vision needs to be integrated into robotic systems in order to automatically analyze and classify sorting objects according to certain characteristics, as well as determine their position and orientation in 3D coordinates. This speeds up production processes and reduces errors, as well as allows the robot to adapt to new working conditions without reprogramming. For robot tasks computer vision can be used to determine the location of objects, as well as the shape, size, color and other characteristics of objects and sort objects based on these characteristics. An important task of computer vision in the context of object sorting tasks is to process the whole working area of the robot.

In a field of robotics, wide-angle cameras have a number of advantages in comparison to conventional cameras[4–6]. A wide angle allows to obtain a more complete information of the location in the work area of the robot, which can be useful for precise positioning and performing complex operations. Instead of installing several conventional cameras, the one single wide-angle camera can be used, which reduces equipment costs and robustness of the system as less elements are used. This simplifies the process of integrating the camera into the robot system and reduces the time it takes to configure it. However, it is not possible to obtain depth

Department of Electric drive, mechatronics and electromechanics, South Ural State University, Chelyabinsk 454080, Russia. ✉email: asp22ct750@susu.ru

information based on a single camera; to eliminate this drawback, a structured light can be integrated into the computer vision system.

The structured light method has its own advantages and provides accurate measurement results[7,8]. This makes it suitable for tasks that require high accuracy, for example in manufacturing[9] or medicine[10]. The structured light method is relatively easy to set up and use in comparison to more complex and expensive technologies such as lidars. Structured light systems can measure distances from a few centimeters to several meters, making them suitable for a variety of tasks. The projector can create a contrasting image even in low light, which allows the structured light method to be used in a variety of conditions. An important point in camera-structured light systems is the process of external calibration of the system, without which the experimental depth measurement results will differ from the real ones[11]. Thus, external calibration plays an important role in ensuring the accuracy and stability operation of computer vision systems.

The contributions of our work are two-fold: (1) A customizable photo-realistic simulator for the CV community working with omnidirectional vision systems with structured light in robotics sorting scenarios. (2) An improved extrinsic calibration of the omnidirectional camera and laser plane is proposed. Results are accurate and robust for simulated and real experiments.

The remainder of this paper is presented in four sections. Section II reviews the related work. Section III provides simulator overview. Section IV is the proposed calibration approach. V presents experiments and error analysis. VI explains robotics sorting system. Finally, the conclusion is drawn in Section VII.

Related work

This paper presents robotics system simulator and an improved extrinsic calibration technique for the omnidirectional vision system with the structured light. The proposed calibration method is tested by means of the simulated and real data. Section II-A presents the simulation environment and reveals its capability. Section II-B is the proposed method for extrinsic calibration between vision sensors.

Simulator

In recent decades, scientists and companies have created a wide range of robotic simulators that facilitate the acquisition of various skills and the testing of theoretical concepts[12]. Since a complete review of all existing simulators is too extensive for this paper, in this section we will focus only on the most relevant works for our research. In the Gazebo[13] and USARSim[14] simulators, it is possible to integrate omnidirectional cameras, structured light, and industrial robots into the environment using certain plugins. In works[15,16] the authors integrated industrial robots into the Gazebo and USARSim software environments. In order to integrate an omnidirectional camera into these simulators[13,14], the authors superimposed images of the environment on the faces of a cube, after which they were able to use this texture to create a hyperbolic mirror or a fisheye camera. However, it was not mentioned how the internal parameters of the camera were obtained, manually or using a calibration process. In[17], the authors integrated an omnidirectional camera and a laser rangefinder into the USARSim environment. However, the authors mentioned that the system parameters were determined manually, and the system calibration process was not used. Also, the process of integrating elements into these software environments using third-party plugins requires certain programming skills, which might be difficult for some users. Using these cameras as default tool settings can be more convenient. In recent years, NVIDIA has introduced a photorealistic robotics simulator known as NVIDIA Isaac Sim[18]. The latest version of this simulator provides support for a fisheye camera. However, this simulator is limited to devices equipped with NVIDIA GPUs. Blender has been considered as a platform for creating photorealistic synthetic images within omnidirectional vision systems in various studies[19,20]. Blender is an open-source toolkit for 3D modeling, rendering, and animation. However, its functionality is not quite suitable for programming tasks and interaction with other software applications, which limits its use in some scenarios. Game engines such as Unity[21] offer significant flexibility as they provide both programming capabilities and realistic graphics and physics. Thus, as part of this work, we decided to release a simulator for the robotics community that can be used on all operating systems and supports the ability to interact with other programs. Cross-platform can be ensured using the Unity game engine, which also supports creation of photorealistic synthetic images and it is free for personal use. Thus, we created a simulator in the Unity software environment with a developed user interface and customizable sections, which does not require specific programming skills from users. To our knowledge, this is the first customizable simulator designed for studying robotic systems that already includes a fisheye camera, which was integrated into the environment based on the work[22] with laser illumination and an industrial robot. The proposed simulator also provides sections for performing internal camera calibration and external system calibration processes based on calibration targets, which we have not seen in other works. To use our simulator, no special skills in Unity are required, since all elements can be controlled through the developed control panel. In addition, if the functionality of the proposed simulator seems insufficient, users can customize the Unity project by adding the appropriate changes and adjustments to it. The processes of integrating new elements into the software environment can be found in the developed documentation.

Calibration of the vision system

In the work[11] the calibration process was shown and analyzed. Authors proposed an approach to calculate the external parameters of the computer vision system including a fisheye camera and a structured light source. The external calibration process was carried out on the basis of the one input image. The calibrated computer vision system was evaluated experimentally using synthetic and real data. The authors demonstrated the advantages of their approach (see Fig. 1B) by comparing it to the conventional calibration method for omnidirectional vision systems, which involves two perpendicular checkerboard patterns (see Fig. 1A). During the experiments, the authors analyzed and compared errors of the extrinsic calibration parameters. The authors demonstrated

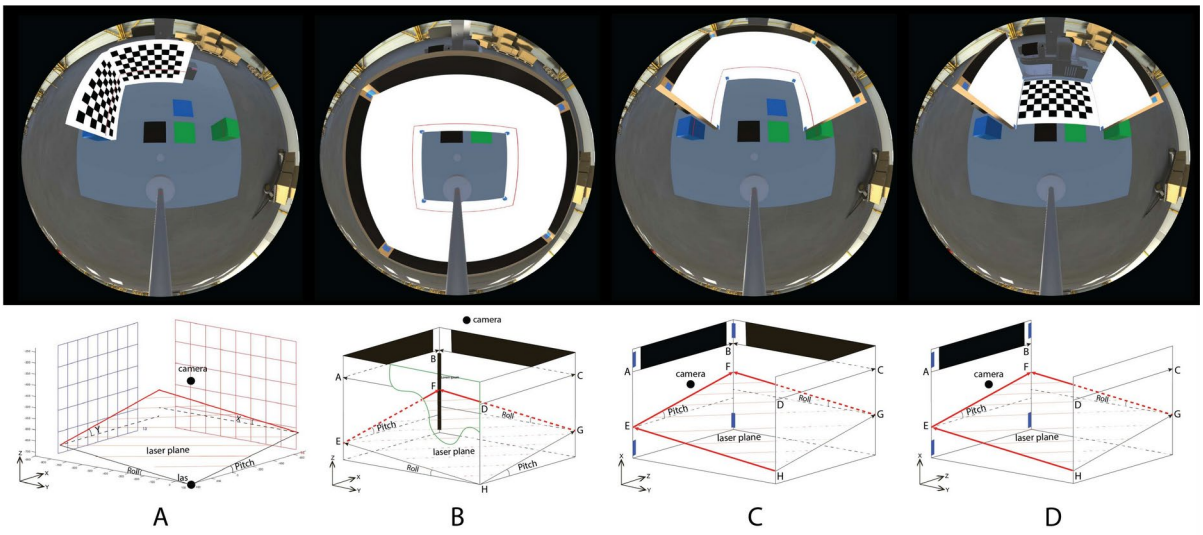

Fig. 1. **A** - commonly used calibration target. **B** - shows calibration target from the work[11]. **C** - shows calibration target from the work[23]. **D** - shows the proposed calibration target. (This figure was created using Unity(version 2022.1.9f1), and then we further processed it using Vector Graphics Software - Adobe Illustrator CC).

that the proposed calibration method is more reliable and achieves higher measurement accuracy. However, the calibration target proposed in this work has limitation to the use. Firstly, the target must be located strictly under the camera in order capture 4 inner sides of the target needed for the calibration. Secondly, camera must be located horizontally, because if it will be located vertically one inner side of the target also will not be visible. Moreover, base holding camera must be moved out of the target in order to do not prevent extracting features of the inner sides of the target need for the calibration process. Taking into consideration these limitations, in the work[23] the authors replaced the calibration target from the work[11]. With the proposed calibration target (see Fig. 1C) authors obtained accurate external parameters of the system. At the same time, by removing one side of the calibration target previous limitations described above were eliminated. However, this target has a rigid configuration, consequently different scale targets need to be developed for the different technological processes.

In this work we propose an optimized calibration target, which includes a smaller number of elements(see Fig. 1D). Because of the less elements in the system the time of mathematical calculations was reduced. The proposed calibration target is more flexible in comparison to previous works[11,23]. By flexible configuration it is possible to adjust sides of the target different technological processes and in order to better evaluate each adjustment with the obtained calibration parameters. This configuration also provides the ability not only to calibrate the computer vision system, but also determine the position of the robot base relative to the camera origin on a single input image.

Simulator overview

Need for simulation

Conducting robotics experiments in a virtual environment has several advantages. There is no need for expensive equipment, materials, and space to conduct research. In a virtual environment, an experiment can be easily repeated with the same initial conditions, which allows testing hypotheses. Virtual laboratories and simulators can be used to train learners without risking safety. Virtual experimentation platforms allow researchers from different countries to work together on common projects, which facilitates international collaboration and knowledge sharing. The virtual environment allows to quickly create and test various versions of robot designs, which speeds up the process of developing and implementing new solutions. Taking into consideration the above problems, we have developed a simulation environment for researchers and engineers in the field of robotics. Our simulation environment provides the ability to conduct experiments in a more cost-effective way compared to real ones. In comparison to physical experiments the simulation environment is characterized by easy setup, high speed of operation and ease of use. The proposed simulator can perform camera and system calibrations, and interact with the robotic complex based on third-party programs.

Individual features

Supported platforms: Windows, Mac OS, and Linux. The simulator includes: (1) interaction with scenes and objects included to simulation environment, (2) communication with other programs by the Transmission Control Protocol (TCP/IP), (3) fully developed applications related to calibration of vision systems and robotics object sorting provide researchers with the opportunity to customize their own experimental installations and develop more advanced algorithms for these tasks. We aimed to create realistic scenes using rendering capabilities such as light sources, reflections, shadows and other spectacular elements. Fig. 2 shows a snapshot of our simulator that demonstrates these rendering capabilities.

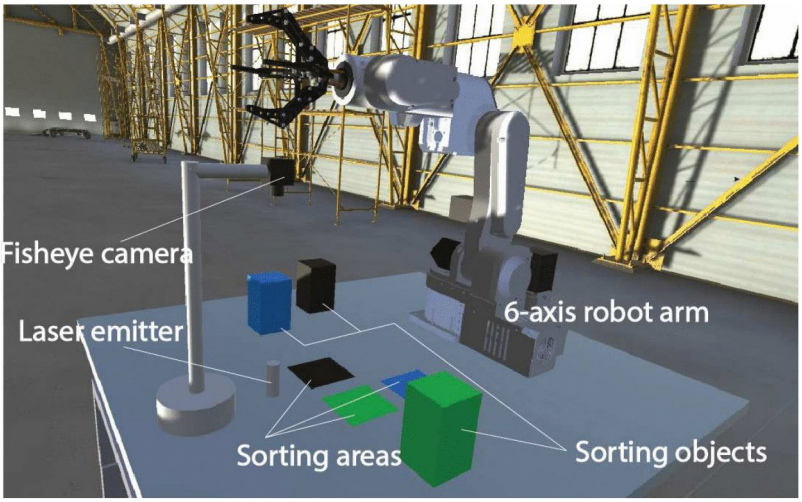

Fig. 2. Main elements of the simulator. (This figure was created using Unity(version 2022.1.9f1), and then we further processed it using Vector Graphics Software - Adobe Illustrator CC).

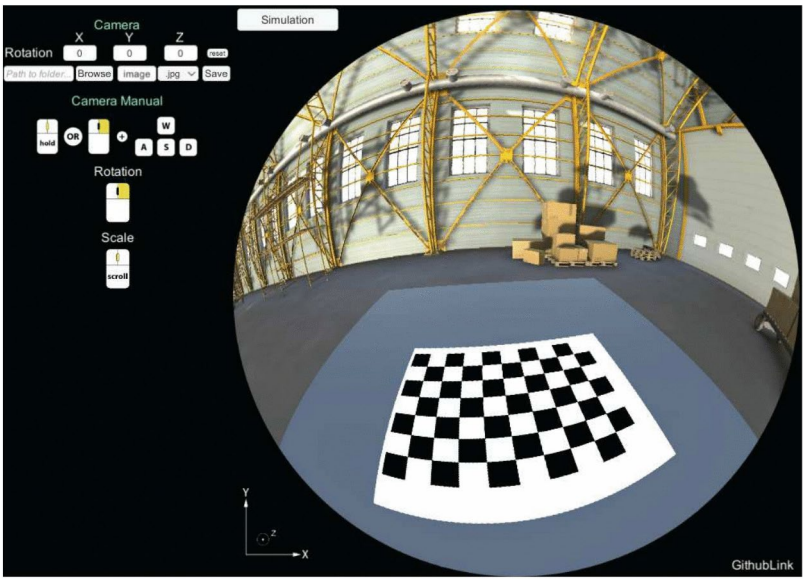

Fig. 3. The calibration screen. (This figure was created using Unity(version 2022.1.9f1)).

Capabilities

The simulator consists of two sections: the camera calibration screen and the main screen. The calibration screen is necessary to collect images of the chessboard in order to calibrate the fisheye camera and obtain its intrinsic parameters. On this screen the position and orientation of the camera can be changed with the mouse in order to take pictures of the chessboard from several angles and save them to a local disk (see Fig. 3). On the main screen, you can configurate elements and control their status on the included to the simulation environment panels located on the left and right (see Fig. 4).

The panel on the left (see zoomed Fig. 5) is responsible for the configuration of the computer vision system and the location of elements in relation to its coordinate system. Camera: change the orientation of the camera and there is also opportunity to save images to a local disk. Laser Plane: activation of the laser, as well as changing the orientation and position in the coordinate system of the camera. Checkerboards: activation of checkerboards, as well as changing their orientation and position in the camera coordinate system. Cubes: activation of sorting objects, for each object its orientation and position are displayed in the camera coordinate system.

The panel on the right (see zoomed Fig. 6) is responsible for the configuration of the six-axis robot arm and the location of the elements in relation to its coordinate system. Robot: displays every joint angle and also gripper position. Cubes: activation of sorting objects, for each object its orientation and position are displayed in the coordinate system of the robot. Target: activation of the proposed calibration target as well as the chessboard

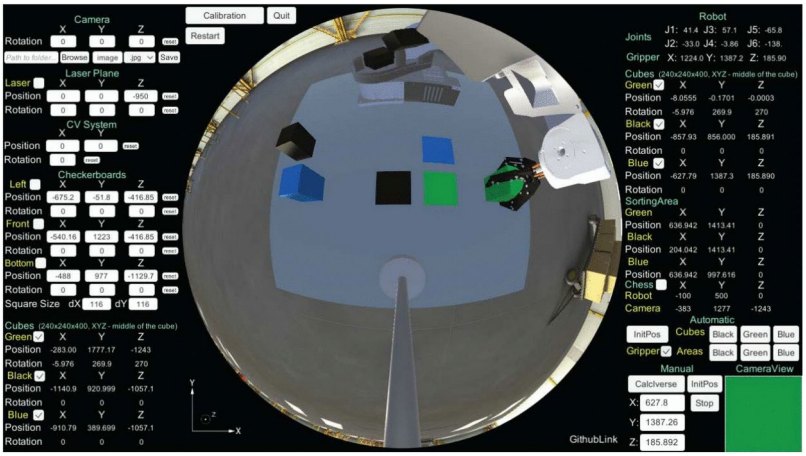

Fig. 4. The main screen. (This figure was created using Unity(version 2022.1.9f1)).

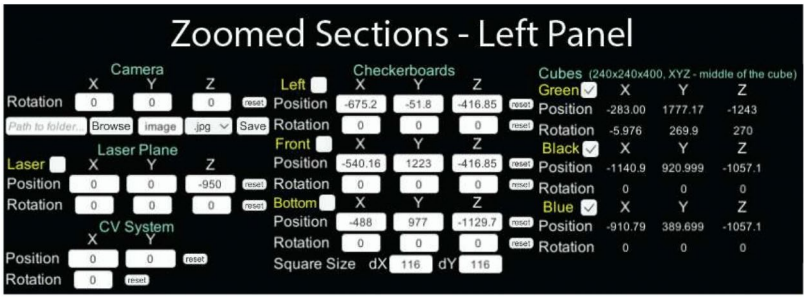

Fig. 5. Zoomed sections of the left panel.

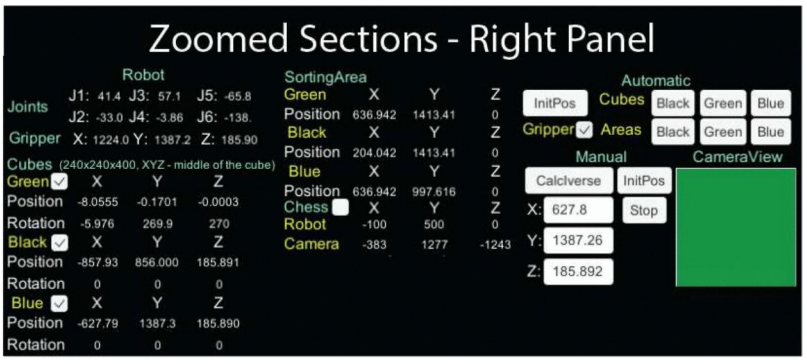

Fig. 6. Zoomed sections of the right panel.

pattern, and position of the target origin is displayed in the coordinate system of the camera and the robot. Automatic: built-in functions for calculating the inverse kinematics to objects and sorting areas and returning the robot to its initial position (coordinates can be changed with program script), as well as activation/deactivation toggle of the end effector of the robot. Manual: three coordinates can be filled into input fields to calculate the inverse kinematics of the robot. In the lower right corner, there is an image from the camera mounted to the end effector of the robot, in this work it is used for the color recognition of objects.

TCP/IP communication makes it possible to operate with the simulator by other programs. For example, we may carry out mapping of the environment to get distance of objects and next we need to provide the end effector of the robot arm with an understating where to move. In this case we can transfer via TCP/IP coordinates to Unity and the build in functions will calculate the inverse kinematics and trajectory for us. Another case is also possible to calculate inverse kinematics and trajectory by other programs and transfer angles via TCP/IP to the

Unity. It is also possible to get images from simulator via TCP/IP, for example from the camera mounted to the end effector of the robot.

Calibration method

System model

The system model was previously described in[11] and just a brief overview is presented in this section. World coordinates of the laser plane (X, Y, Z) can be obtained as follows:

$$\begin{bmatrix} u \\ v \\ f\,(p) \end{bmatrix} \times \begin{bmatrix} r_1^c & r_2^c & r_3^c \end{bmatrix} \begin{bmatrix} r_1^l & r_2^l & r_3^l & t^l \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \tag{1}$$

where u, v represent pixel coordinates;r_1^c, r_2^c, r_3^c, r_1^l, r_2^l, r_3^l, r^l represent column vectors of the rotation matrix of the camera and transformation matrix of the laser plane respectively. The polynomial $f(p)$ has the following form:

$$f\,(p) = a_0 + a_2 p^2 + \cdots + a_N p^N \tag{2}$$

$$p = \sqrt{(u - u_c)^2 + (v - v_c)^2} \tag{3}$$

where a_i are coefficients; N the degree; point u_c, v_c is the center of the image.

The laser emitter has a rigid configuration, consequently distance to the camera is constant: along the Z-axis. Taking into account the aforementioned criteria, for the proposed vision system Eq. (1) can be simplified as:

$$\begin{bmatrix} u \\ v \\ f\,(p) \end{bmatrix} \times \begin{bmatrix} r_1^c & r_2^c & r_3^c \end{bmatrix} \begin{bmatrix} r_1^l & r_2^l & t^l \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0 \tag{4}$$

Problem

In[23] was presented a novel calibration technique for obtaining extrinsic parameters between the camera and laser plane. This calibration method was based on the use of a three-sides target and has proven its accuracy and reliability in comparison with other calibration methods. In this section, we propose the optimized calibration target and evaluate its robustness and performance compared to previous calibration methods. As can be seen from Fig. 7b, the optimized calibration target consists of two sides, the calibration target from work[23] consists of three sides (see Fig. 7a). We also placed a chessboard between the sides of the target in order to not only calibrate the computer vision system, but also based on a single input image determine the position of the robot in relation to the camera coordinate system.

Calibration procedure

The goal of the extrinsic calibration is to find parameters of the rotation matrix of the camera and transformation matrix of the laser plane respectively. In general form this optimization problem can be formulated as follows:

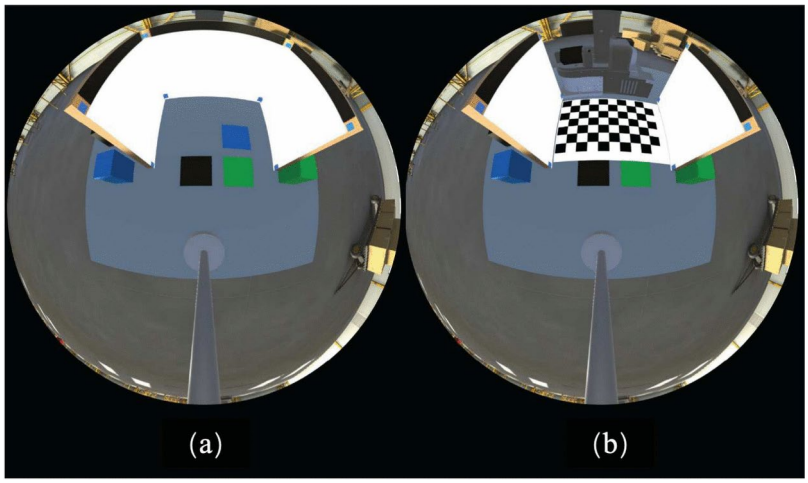

Fig. 7. (**a**) shows previous configuration. (**b**) shows proposed configuration. (This figure was created using Unity(version 2022.1.9f1)).

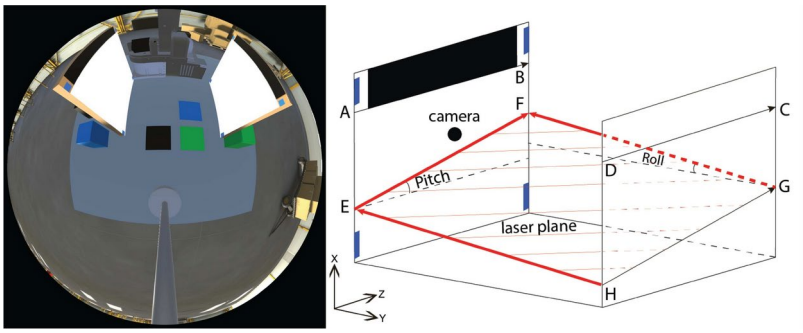

Fig. 8. The proposed calibration target. (This figure was created using Unity(version 2022.1.9f1), and then we further processed it using Vector Graphics Software - Adobe Illustrator CC).

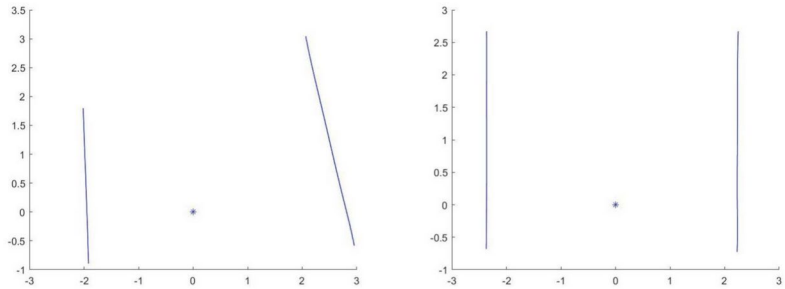

Fig. 9. (**a**) projection with unkown pitch and roll. (**b**) projection with kown pitch and roll.

$$\begin{cases} min_{R^c,R^l,T^l} \| f(R^c, \left[R^l, T^l\right]) \|^2 \\ subject\ to\ f(R^c, \left[R^l, T^l\right]) = \begin{bmatrix} u \\ v \\ f(p) \end{bmatrix} \times R^c \left[R^l \mid T^l\right] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \\ R^c = [r_1^c\ r_2^c\ r_3^c] \\ \left[R^l \mid T^l\right] = \left[r_1^l\ r_2^l\ t^l\right] \end{cases} \tag{5}$$

where R^c represents the rotation matrix of the camera and $\left[R^l \mid T^l\right]$ represent transformation matrix of the laser plane, consisting of rotation R^l and translation T^l parts respectively.

An improved calibration target was developed (see Fig. 8) for solving the optimization problem described in Eq. (5). The main advantage of this target is its versatility, as it can be applied to various configurations of a vision system, as well as its flexibility, as it can be simply placed in front of the mobile robot. The proposed target allows an extrinsic calibration to be performed by only capturing a single snapshot, this procedure is explained below.

Extrinsic calibration of the vision system

This section explains the process of obtaining parameters forming the camera rotation matrix R^c as described in Eq. (5). In order to carry out the calibration process to obtain camera extrinsic parameters, first of all, pixel coordinates belonging to the border (between white and black regions) of the target are projected by Eq. (4) to the world coordinate system (see Fig. 9a). After that for every parameter, namely for the pitch, roll, and yaw, the optimization is described as a series of Eqs. (6), 7), (8). The pitch is calculated when projected to the world coordinates vectors $\overline{AB}$ and $\overline{DC}$ of the target collinear to each other. This optimization problem can be formulated as follows:

$$\begin{cases} min_{pitch} \|f(pitch)\|^2 \\ subject\ to\ f(pitch) = \frac{\overline{AB}_Y}{\overline{AB}_Z} - \frac{\overline{DC}_Y}{\overline{DC}_Z} \end{cases} \tag{6}$$

The yaw is calculated when projected to the world coordinates vectors $\overline{AB}$ and $\overline{DC}$ of the target are not rotated around X-axis. This optimization problem can be formulated as follows:

$$\begin{cases} min_{yaw} \left\| f(yaw) \right\|^2 \\ subject\ to\ f(yaw) = \frac{\overline{AB}_Y}{\overline{AB}_Z} \cdot \frac{\overline{DC}_Y}{\overline{DC}_Z} \end{cases} \tag{7}$$

Once the pitch and yaw are known it is possible to calculate the roll. Sides of the calibration target are the same length as well as the extracted border between black and white regions also is the same length. The roll can be found by minimization the difference between length of vectors $\overline{AB}$ and $\overline{DC}$ whereas pitch and yaw obtained during previous steps are constant. This minimization problem can be written as follows:

$$\begin{cases} min_{roll} \left\| f(roll) \right\|^2 \\ subject\ to\ f(roll) = \left| \overline{AB} \right| - \left| \overline{DC} \right| \end{cases} \tag{8}$$

At this point, pixel coordinates belonging to the border of the target can be projected by Eq. (4) to the world ones with the pitch, roll, and yaw, determined during the calibration procedure (see Fig. 9b). Once the camera is calibrated, we can move to the calibration of the laser plane.

Extrinsic calibration of the laser plane

This section outlines the process of obtaining parameters forming the transformation matrix $\left[R^l \mid T^l \right]$ of the laser plane, which are part of the Eq. (5), whereas parameters of R^c are known and constant. First of all, extracted pixel coordinates of the laser beam are projected by Eq. (4) to the world ones (see Fig. 10a). After that for every parameter related with the transformation matrix of the laser plane, the minimization problem is formulated by a series of Eqs. (9), (10), (11).

The pitch is calculated when projected to the world coordinates vectors $\overline{EF}$ and $\overline{HG}$ of the target collinear to each other. This optimization problem can be formulated as follows:

$$\begin{cases} min_{pitch} \left\| f(pitch) \right\|^2 \\ subject\ to\ f(pitch) = \frac{\overline{EF}_Y}{\overline{HG}_Z} - \frac{\overline{EF}_Y}{\overline{HG}_Z} \end{cases} \tag{9}$$

Another parameter related with the R^l is the roll. Sides of the calibration target are the same length as well as the extracted laser strip is also the same length. The roll can be found by minimization the difference between length of vectors ($\overline{EF}$ and $\overline{HG}$ whereas pitch and yaw obtained during previous steps are constant. This minimization problem can be written as follows:

$$\begin{cases} min_{roll} \left\| f(roll) \right\|^2 \\ subject\ to\ f(roll) = \left| \overline{EF} \right| - \left| \overline{HG} \right| \end{cases} \tag{10}$$

Finally, the last unknown parameter included to the transformation matrix of the laser plane representing the distance to the laser plane can be calculated. The real distance D_1 between the left and right sides of the target is known. The distance D_2 between sides of the target can be found experimentally from the world coordinates of the laser. Thus, the depending variable representing the distance to the laser plane can be calculated as the difference between D_1 and D_2. This minimization problem can be written as follows:

$$\begin{cases} min_{dist} \left\| f(dist) \right\|^2 \\ subject\ to\ f(dist) = D_1 - D_2 \\ D_2 = (\frac{Y_H + Y_G}{2} - \frac{Y_E + Y_F}{2}) \end{cases} \tag{11}$$

Afterwards, pixel coordinates of the laser beam can be projected by Eq. (4) to the world ones with the pitch, roll, and distance to the laser plane, determined during the extrinsic calibration (see Fig. 10b).

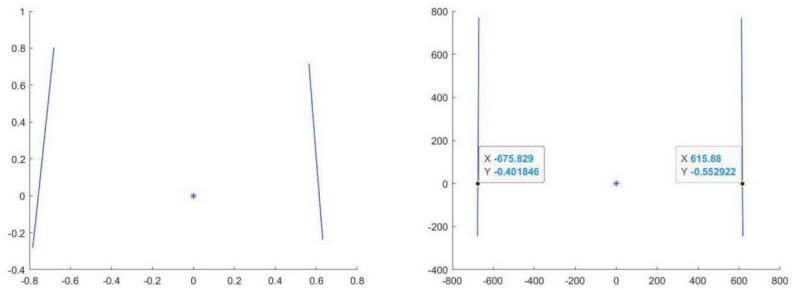

Fig. 10. (**a**) projection with unkown pitch, roll and distance. (**b**) projection with kown pitch, roll and distance.

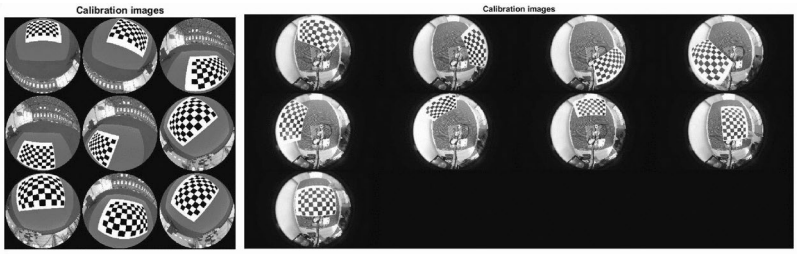

Fig. 11. Left image shows images from the simulation environment. Right image shows images from the real environment. (The left image was created using Unity(version 2022.1.9f1), and the right image was captured using HF890 industrial fisheye camera).

	Simulation Environment	Real Environment
Image resolution, pixels	1920x1920	1920x1080
Checkerboard patterns		
Pattern size	9x6	9x6
Square size, mm	116x116	26x26

Table 1. The configurations of the simulation and real environments.

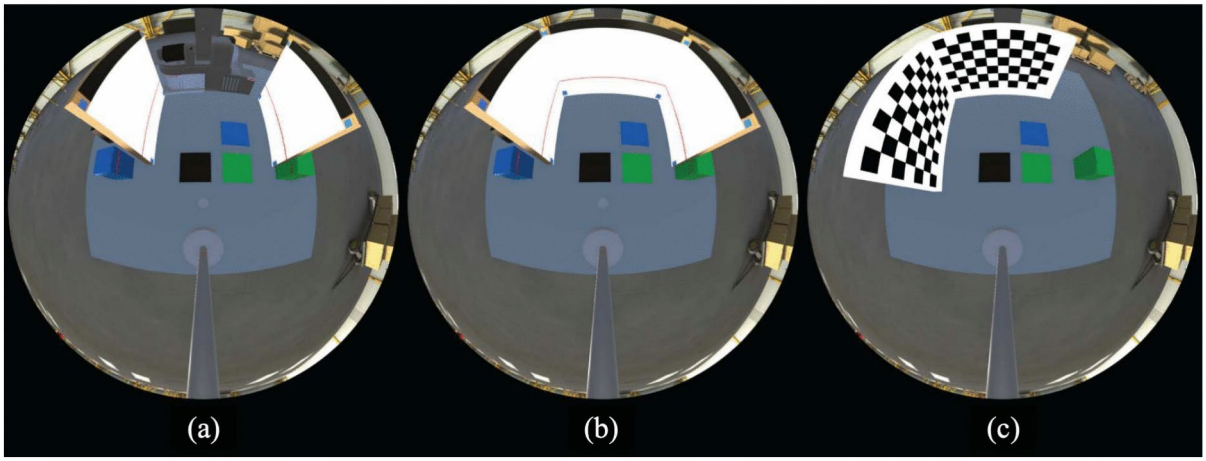

Fig. 12. Configuration of the vision system. (**a**) Proposed. (**b**) Method 1. (**d**) Method 2. (This figure was created using Unity(version 2022.1.9f1)).

Vison system calibration

Intrinsic parameters of the camera

This subsection explains how the intrinsic camera parameters were estimated. For calibration procedure we followed a similar approach as the one introduced in[24] by D. Scaramuzza. Nine images were captured with the simulator and with the real fisheye camera as shown in Fig. 11. Once these images were captured, the calibration process was carried out by means of a Toolbox extension[25]. The configurations of the simulation and real environments are shown in the Table 1.

Extrinsic parameters of the vision system

Experimental setup

For evaluating the merits of the proposed calibration technique (see Fig. 12a), as well as for demonstrating the quality of its performance, it was compared with two other calibration methods. The experimental setup is similar to the one presented in[23] and depicted in Table 2. Method 1 is based on the three-sides target (see Fig. 12b) and its calibration technique was considered in[23]. Method 2 is based on the checkerboard (see Fig. 12 c), and its calibration technique was considered in[26]. Thirty-five configurations of the vision system were included in the experiment, where parameters of the rotation matrices of the camera and laser plane (R^c, R^l, T^l) were varied but do not exceed ten degrees.

Representation of the Target	
Proposed	Target with 2 sides
Method 1	Target with 3 sides
Method 2	Target with Checkerboards
Features of the target	
Proposed	Size of the target is 1084.5x1084.5 mm
Method 1	Distance to the left side of the target is 646 mm and to the front side is 746 mm
Method 2	Pattern has 9x6 squares. Each square is 97x97 mm
Image resolution	
Proposed	1920x1920 pixels
Method 1	
Method 2	

Table 2. The experimental setup.

	Rotation Matrix of the Camera			*Transformation Matrix of the Laser Parameters*		
	Pitch, deg	**Roll, deg**	**Yaw, deg**	**Pitch, deg**	**Roll, deg**	**Yaw,deg**
MAE						
Proposed	0.05	0.13	0.06	0.14	0.09	2.79
Method 1	0.06	0.12	0.06	0.16	0.12	2.81
Method 2	0.11	0.15	0.09	0.29	0.08	3.32
RMSE						
Proposed	0.05	0.13	0.07	0.16	0.10	2.92
Method 1	0.06	0.12	0.07	0.18	0.12	2.97
Method 2	0.13	0.16	0.11	0.37	0.23	3.82
Average calibration time,seconds						
Proposed	28.39					
Method 1	33.99					
Method 2	25.41					

Table 3. The experiment results.

Results and dicussion

Table 3 presents the mean absolute error (MAE) and the root mean squared error (RMSE) for the comparative analysis between the calibration methods. The proposed method showed almost the same results as the method 1, and these two methods outperforms the method 2. It is also worthwhile mentioning that the proposed method is faster than method 1 and method 2.

In work[23], it was shown that the calibration Method 1 is outperforms to Method 2. The purpose of our work was to prove previous results, and also to show that by reducing the number of sides of the calibration target from three sides (Method 1) to two sides (the Proposed Method), we will be able to get similar results of external parameters, while increasing the camputational speed of the calibration process, which we we can see from Table 3. Moreover, the third side of the target was removed in order to provide the flexibility of the calibration process in relation to the computer vision system calibration and obtaining the coordinates of the robot manipulator in the camera coordinate system. Because of the two sides of the target, it is possible to place a chessboard pattern in between and measure the distance between the robot and the chessboard pattern origin. Calibration in this case is performed on the basis of the one single input image, a description of this process is given in Section 5.

Real data

Experimental setup

In order to show that the proposed calibration approach can be applied to real scenarios, we created real experiment setup similar to the simulation environment(see Fig. 13a). For the simple testing purpose, the real calibration target was created from a cardboard box (see Fig. 13b). The distance between sides of the calibration target was equal to 350 mm. The image resolution of the fisheye camera is equal to 1920x1080 pixels. The laser emitter with a wavelength of 650 nm was chosen. The distances from camera to the three obstacles was measured by the dimensional board. The configuration of the real environment is depicted in the Table 1.

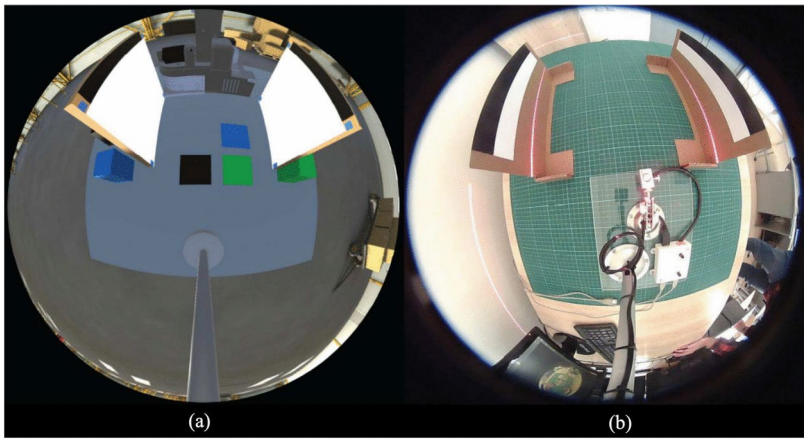

Fig. 13. (**a**) shows simulation target. (**b**) shows real target. (Figure(**a**) was created using Unity(version 2022.1.9f1), and Figure(**b**) was captured using HF890 industrial fisheye camera).

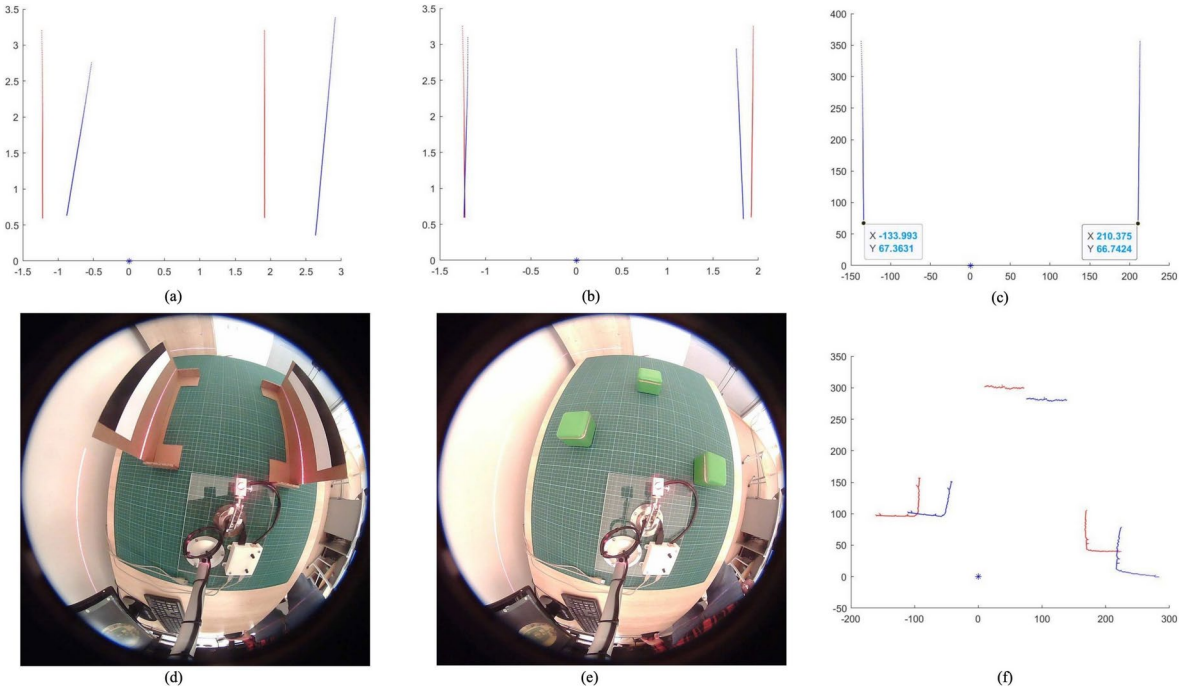

Fig. 14. Extrinsic calibration and mapping in the real environment. (**a**) Blue - initial projection [0;0;0]. Red - projection with the optimized camera orientation [1.2;-6.5;-8.4]. (**b**) Blue - initial projection [0;0;0]. Red - projection with the optimized laser orientation [-1.2;1.6;0]. (**c**) Optimized laser distance - 213 mm. Distance between sides is of calibration target equal to 350 mm. (**d**) Input image for extrinsic calibration. (**e**) Input image for mapping. (**f**) Blue - mapping of the not calibrated system. Red - mapping of the calibrated system. (Figure(**a**)(**b**)(**c**)(**f**) were created using MATLAB R2022a, and Figure(**d**)(**e**) were captured using HF890 industrial fisheye camera).

Results and dicussion

In the Fig. 14 is displayed the visual representation of the extrinsic calibration of the vision system and mapping results in the real environment. First of all, extrinsic parameters of the vision system were found by the input image, which is shown in the Fig. 14d. Fig. 14a shows the process of obtaining camera parameters, namely pitch, roll, and yaw. This difference between zero camera angles and angles found by calibration process is quite significant. Fig. 14b shows the process of obtaining laser plane parameters, namely pitch, and roll. Projecting laser points to the world coordinates should give us parallel lines as sides of the calibration target are parallel to each other. In this case projection with angles found by calibration process provides us better result. Fig. 14c shows us the projection of the laser points with the distance between camera origin and origin of the laser plane. This distance was found during the calibration procedure.

	Front, mm	Right, mm	Left, mm
Real	299.0	160.0	90.0
Experiment	301.5	165.2	93.4
AE	2.5	5.2	3.4

Table 4. The evaluation of the mapping results.

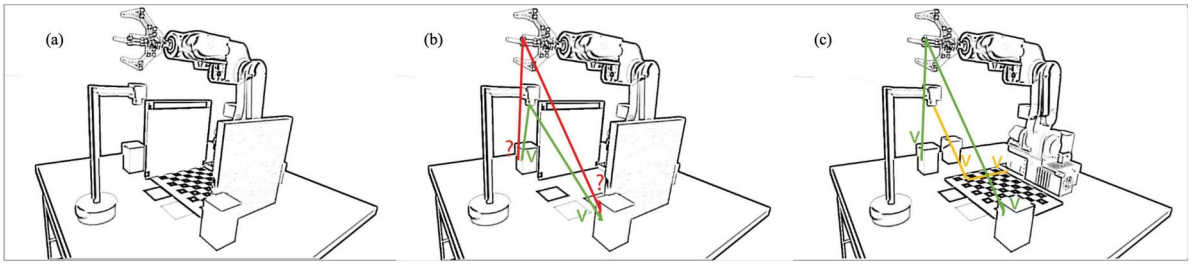

Fig. 15. Robotics system calibration. (**a**) Calibration target. (**b**) Part of the calibration target to get extrinsic parameters of the vision system. (**c**) Part of the calibration target to get the robot position in camera coordinate system.

Once all of the extrinsic parameters were found we can also verify our vision system by mapping results to the objects placed to the environment. The input image for mapping procedure is shown in the Fig. 14e. Fig. 14f shows mapping results obtained by the calibrated vision system and compared to the not calibrated case. This figure shows us importance of the vision system calibration as by knowing its extrinsic parameters it is possible to obtain the actual location and orientation of objects in the environment. Table 4 shows us average error between real position of objects and position obtained during experiments for the calibrated vision system. This error is not significant considering the fact that for calibration process we used a simple cardboard box.

The main aim of this subsection was to show that the proposed calibration approach has a practical application. The experiments carried out in the proposed simulation environment significantly facilitate the understanding of how to perform real experiments. We should also mention that some of the algorithms had to be modified. Fig. 13 shows difference between images capture by simulation and real environments. Firstly, for the border extraction between black and white regions the synthetic image has a better quality and therefore more clear pixel representation of black and white pixels. In case of the real capture, we increased the range of searching areas of these pixels. As we increased this searching range some of the wrong found borders had to be eliminated. For example, the area between wall and black background of the image was also detected as a border which we eliminated in a program way. Secondly, the laser beam in the simulation environment has a constant illumination, therefore it can be extracted from the image by simple threshold algorithms. In the real case laser illumination has different distribution within image range and more complex algorithms of laser beam extraction should be used in this case[27].

Robotics sorting system

Once the vision system is calibrated, we can proceed to the objects sorting procedure. But before that, we need to make a relationship between coordinate system of the camera and the coordinate system of the robot arm. For this purpose, a chessboard pattern was added to the calibration target (see Fig. 7b). These steps are described in details below.

Coordinate system of the robot arm

Fig. 7b and Fig. 15a shows the calibration target by which we can calibrate the computer vision system and get the coordinates of the robot arm. In Section 5 we demonstrated the process of obtaining extrinsic parameters of the vision system by the part of the calibration target which is shown in the Fig. 15b and obtained distances to objects by mapping. Therefore, we know position and orientation of objects in the camera coordinate system, but in the robot coordinate system its currently unknown. Fig. 15c shows the part of the calibration target which is used in order to coordinates of the robot arm after that it is possible to operate with objects by the end effector of the robot arm. Therefore, by attaching checkerboard pattern to the robot base at the know distance and getting checkerboard coordinates by camera we can get position and orientation of the robot arm in the camera coordinate system (see Fig. 16).

Experimental setup

Firstly, by the proposed calibration target (see Fig. 7b and Fig. 15a) the vision system was calibrated and coordinates of the robot arm in the coordinate system of the camera were obtained. Secondly, for the three placed objects to the environment we carried out mapping and also displayed location of the camera and robot arm on the map. Lastly, the robot arm was performed sorting object procedure based on the obtained position

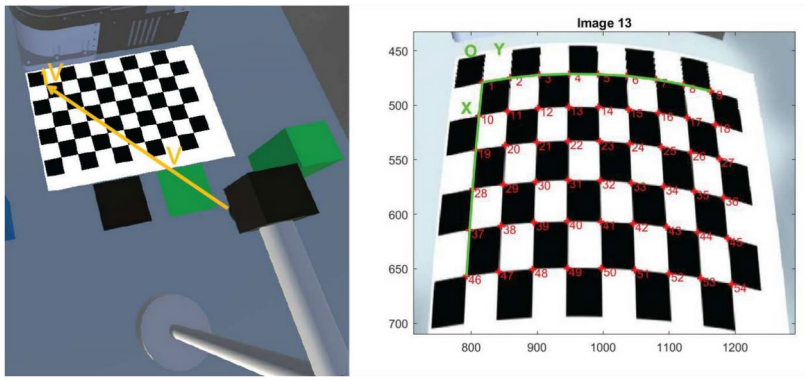

Fig. 16. Coordinates of the robot arm in the coordinate system of the camera. (The left image was created using Unity(version 2022.1.9f1), the right image was created using MATLAB R2022a, and then we further processed it using Vector Graphics Software - Adobe Illustrator CC).

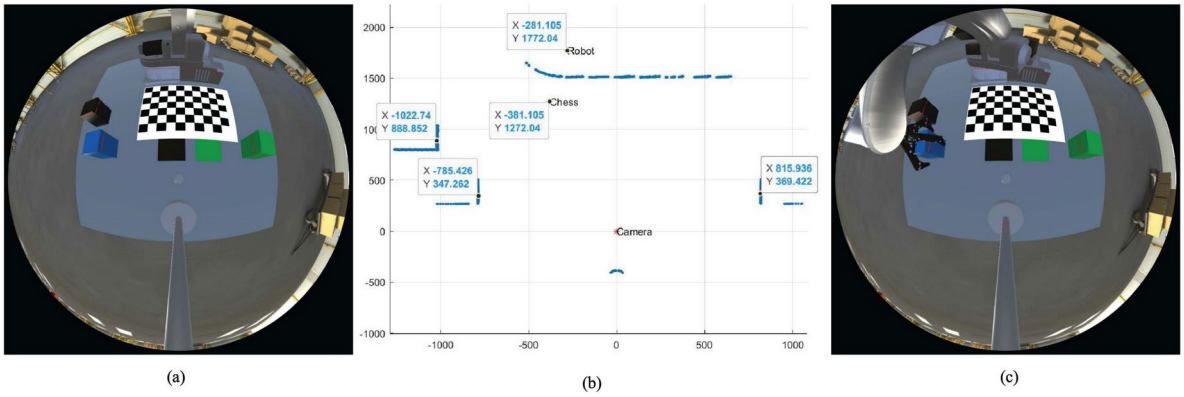

Fig. 17. Input image for mapping. (**a**) Coordinate systems of the camera and robot arm as well as objects location. (**b**) Inverse kinematics calculation based on the object coordinates. (Figure(**a**)(**c**) were created using Unity(version 2022.1.9f1), Figure(**b**) was created using MATLAB R2022a, and then we further processed it using Vector Graphics Software - Adobe Illustrator CC).

	Checkerboard origin		Objects position		
	X , mm	Y , mm	Black, mm	Blue, mm	Green, mm
Real	383.00	1277.00	1140.90	910.82	941.08
Experiment	381.10	1272.04	1142.74	905.43	935.94
AE	1.90	4.96	1.84	5.39	4.86

Table 5. The evaluation of the mapping results.

and orientation for each object. For the testing purpose we used geometrical approach[28] to calculate the inverse kinematics of the 6-axis robot arm and for each of the joint we used a simple linear trajectory between two points in world coordinates. TCP/IP connection was used in order to communicate between Matlab and Unity programs (see Section 3C for more details).

Results and dicussion

Fig. 17a is the original image captured, showing the coordinate system of the camera and the robot arm, as well as the position of the object. Fig. 17b shows the visual representation of the mapping results, where also displayed obtained origin of the checkerboard pattern and calculated origin of the robot arm. Based on this it was possible to calculate position of every object in the coordinate system of the robot arm and based on these coordinates calculate inverse kinematics to approach objects (see Fig. 17c). In the Table 5 is show error analysis. It can be seen that position of the checkerboard pattern and position of objects were obtained quite accurate with the calibrated vision system. The small misalignment in the object position can be neglected by the gripper size.

This section demonstrated the compatibility of the proposed calibration target. However, the process of obtaining the robot arm position by placing the checkerboard pattern at know distance from the robot is not new, the main goal was to show the flexibility of the proposed calibration target. Therefore, by one single input image it was possible to calibrate the vision system and to get coordinates of the robot arm.

Conclusion and future work

This paper proposed a photorealistic simulator that can be used for calibrating and sorting tasks in the field of robotics. With the aid of the developed simulator, two practical applications were investigated: extrinsic calibration of the vision system and sorting of objects by the six-axis robot manipulator. As far as we know, this is the first simulator that includes these elements and modes of operation by default. Therefore, this simulator might be useful for researchers and help them conduct experiments in a safety way before moving to the real setup.

As for the proposed calibration method, it is worthwhile mentioning that the processes itself is quite simple as only one input image is required to implement the extrinsic calibration of the system. The simulation results demonstrated that our calibration method is outperforms existing modern calibration approaches, and is also applicable to real systems. However, while transferring algorithms from synthetic images to real ones, we had to adjust the image processing algorithms. As it was discussed earlier, we faced these issues when the border between black and white pixels was extracted and also for extracting the laser beam from the image. Therefore, further work can be focused on the unification of the proposed algorithms for virtual and real environments.

Data availability

The data sets generated during and/or analysed during the current study are not publicly available for privacy reasons, but are available from the corresponding author on reasonable request.

Received: 21 October 2024; Accepted: 17 March 2025
Published online: 25 March 2025

References

1. Xue, Y., Qiu, Z., Zhang, H., Wang, Z. & Hirai, S. A synthetic dataset for robotic food handling system. In *2024 IEEE/SICE International Symposium on System Integration (SII)*, 410–415, https://doi.org/10.1109/SII58957.2024.10417293 (2024).
2. Naito, K., Shirai, A., Kaneko, S.-i. & Capi, G. Recycling of printed circuit boards by robot manipulator: A deep learning approach. In *2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 1–5, https://doi.org/10.1109/ROSE52750.2021.9611773 (2021).
3. Yan, S., Lin, H. & Yao, L. Design of automatic drug sorting system in warehouse based on 3d camera and cooperative robot. In *2021 9th International Conference on Orange Technology (ICOT)*, 1–4, https://doi.org/10.1109/ICOT54518.2021.9680612 (2021).
4. Deshpande, K., Heindl, C., Stübl, G., Kollingbaum, M. J. & Pichler, A. Novel first person view for human 3d pose estimation in robotic applications using fisheye cameras. In *2024 10th International Conference on Automation, Robotics and Applications (ICARA)*, 112–116, https://doi.org/10.1109/ICARA60736.2024.10553148 (2024).
5. Gu, Z., Liu, H. & Zhang, G. Real-time indoor localization of service robots using fisheye camera and laser pointers. In *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, 1410–1414, https://doi.org/10.1109/ROBIO.2014.7090531 (2014).
6. Zhang, Z., Rebecq, H., Forster, C. & Scaramuzza, D. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 801–808, https://doi.org/10.1109/ICRA.2016.7487210 (2016).
7. Lu, J., Zhang, J., Ye, M. & Mi, H. Review of the calibration of a structured light system. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 4782–4786, https://doi.org/10.1109/IECON43393.2020.9255321 (2020).
8. Gao, S., Zhao, M., Zhang, L. & Zou, Y. Dual-beam structured light vision system for 3d coordinates measurement. In *2008 7th World Congress on Intelligent Control and Automation*, 3687–3691, https://doi.org/10.1109/WCICA.2008.4593515 (2008).
9. Cao, Z., Zhong, F. & Wu, Q. Size measurement based on structured light sensor. In *2011 Second International Conference on Digital Manufacturing & Automation*, 1098–1101, https://doi.org/10.1109/ICDMA.2011.270 (2011).
10. Olesen, O. V., Paulsen, R. R., Hojgaard, L., Roed, B. & Larsen, R. Motion tracking for medical imaging: a nonvisible structured light tracking approach. *IEEE transactions on medical imaging* **31**, 79–87. https://doi.org/10.1109/TMI.2011.2165157 (2011).
11. Kholodilin, I., Li, Y. & Wang, Q. Omnidirectional vision system with laser illumination in a flexible configuration and its calibration by one single snapshot. *IEEE Transactions on Instrumentation and Measurement* **69**, 9105–9118. https://doi.org/10.1109/TIM.2020.2998598 (2020).
12. Collins, J., Chand, S., Vanderkop, A. & Howard, D. A review of physics simulators for robotic applications. *IEEE Access* **9**, 51416–51431. https://doi.org/10.1109/ACCESS.2021.3068769 (2021).
13. Koenig, N. & Howard, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3, 2149–2154, https://doi.org/10.1109/IROS.2004.1389727 (2004).
14. Wang, J., Lewis, M. & Gennari, J. Usar: A game based simulation for teleoperation. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting* **47**, 493–497. https://doi.org/10.1177/154193120304700351 (2003).
15. Noh, S., Park, C. & Park, J. Position-based visual servoing of multiple robotic manipulators: Verification in gazebo simulator. In *2020 international conference on information and communication technology convergence (ICTC)*, 843–846, https://doi.org/10.1109/ICTC49870.2020.9289554 (2020).
16. Okamoto, S., Kurose, K., Saga, S., Ohno, K. & Tadokoro, S. Validation of simulated robots with realistically modeled dimensions and mass in usarsim. In *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, 77–82, https://doi.org/10.1109/SSRR.2008.4745881 (2008).
17. Nguyen, Q., Visser, A. *et al.* A color based rangefinder for an omnidirectional camera. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2009), Workshop on Robots, Games, and Research: Success stories in USARSim, IEEE*, 41–48 (2009).
18. Nvidia isaac sim [online]. *Available:* https://developer.nvidia.com/isaac-sim.
19. Won, C., Ryu, J. & Lim, J. Sweepnet: Wide-baseline omnidirectional depth estimation. In *2019 International Conference on Robotics and Automation (ICRA)*, 6073–6079, https://doi.org/10.1109/ICRA.2019.8793823 (2019).
20. Zhang, Z., Rebecq, H., Forster, C. & Scaramuzza, D. Benefit of large field-of-view cameras for visual odometry. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 801–808, https://doi.org/10.1109/ICRA.2016.7487210 (2016).
21. Unity [online]. *Available:* https://unity.com.

22. Bourke, P. Creating fisheye image sequences with unity3d. *Available:* https://www.researchgate.net/publication/279963195_Creating_fisheye_image_sequences_with_Unity3D (2015).
23. Kholodilin, I., Li, Y., Wang, Q. & Bourke, P. D. Calibration and three-dimensional reconstruction with a photorealistic simulator based on the omnidirectional vision system. *International Journal of Advanced Robotic Systems* **18**, 17298814211059312. https://doi.org/10.1177/17298814211059313 (2021).
24. Scaramuzza, D., Martinelli, A. & Siegwart, R. A toolbox for easily calibrating omnidirectional cameras. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5695–5701, https://doi.org/10.1109/IROS.2006.282372 (2006).
25. Urban, S., Leitloff, J. & Hinz, S. Improved wide-angle, fisheye and omnidirectional camera calibration. *ISPRS Journal of Photogrammetry and Remote Sensing* **108**, 72–79. https://doi.org/10.1016/j.isprsjprs.2015.06.005 (2015).
26. Xu, J., Gao, B., Liu, C., Wang, P. & Gao, S. An omnidirectional 3d sensor with line laser scanning. *Optics and Lasers in Engineering* **84**, 96–104. https://doi.org/10.1016/j.optlaseng.2016.04.001 (2016).
27. Zou, Y., Cai, S., Li, P. & Zuo, K. Features extraction of butt joint for tailored blank laser welding based on three-line stripe laser vision sensor. In *2017 29th Chinese Control And Decision Conference (CCDC)*, 7736–7739, https://doi.org/10.1109/CCDC.2017.7978594 (2017).
28. Neppalli, S., Csencsits, M. A., Jones, B. A. & Walker, I. A geometrical approach to inverse kinematics for continuum manipulators. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3565–3570, https://doi.org/10.1109/IROS.2008.4651125 (2008).

Author contributions

I.K and M.G proposed a concept of the manuscript. I.K, ZH.Z designed the experiments. I.K and QH.G executed the experiments. I.K and ZH.Z performed the statistical analysis of the data. I.K, ZH.Z and QH.G drafted the manuscript. M.G reviewed and edited the manuscript.All authors read and approved the final manuscript. None of the authors have any competing interests.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Z.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025