Depth Camera Calibration Using 4 Spheres on Tetrahedron Corners

Esra Tunçer
Electrical and Electronics Engineering
Izmir Institute of Technology
Izmir, Turkey
tunceresra91@gmail.com
ORCID: 0000-0001-5027-2408

Şevket Gümüştekin
Electrical and Electronics Engineering
Izmir Institute of Technology
Izmir, Turkey
sevketgumustekin@iyte.edu.tr

2022 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-8894-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ASYU56188.2022.9925282

Abstract— **In this study, a system including 4 depth cameras (Kinect v1) was constructed for 3-dimensional reconstruction of an object. As with all cameras, the calibration parameters of the cameras used must be known in order to transform from image coordinates to world coordinates in depth cameras. Contrary to the studies that use flat calibration objects such as checkerboard and infrared images, which are frequently encountered in the literature, in this study, only depth images of the cameras were used to find intrinsic and extrinsic parameters of a depth camera. For the calibration process, a new tetrahedron calibration object with 4 equiradius spheres was designed, the sphere centers were determined in the world and image coordinates, and the projection matrix method was applied. The minimum number of matching points required by the original projection matrix method was reduced from 6 to 4 with the help of depth information. A new calibration object and method was developed to provide successful 3-dimensional reconstruction results of 4 depth cameras with the use of only 4 matching points and can be used in all types of depth cameras.**

Keywords— depth camera calibration; projection matrix method; multiple Kinect; 3-dimensional reconstruction.

I. Introduction

A depth camera is a system that uses physical properties of light to measure the distance between an object and a camera. Two different approaches are commonly used: structured light and time of flight. In time of flight approach the traveling time of the light is calculated and in structured light technology, which is used in the depth camera of Kinect v1, an infrared pattern is sent and the disparity between sent and received light pattern is calculated to find the distance of an object. RGBD cameras that can keep the color and distance information of the object at the same time are used in many studies, some use both information, while some studies use only the depth camera part of these RGBD cameras, as in our study.

Finding the intrinsic and extrinsic calibration parameters of the cameras is a necessary process to obtain information about the positions of the objects in 3-dimensional space. The most commonly used model to obtain these parameters in the field of computer vision is the pinhole camera model and the use of images taken from different directions of the checkerboard.[1] The most widely used tool that performs this operation is Bouguet's toolbox.[2]

Pinhole camera model can be used to find the calibration parameters (focal length, optical center) of depth cameras. As with RGB camera studies, checkerboard is also widely used in the depth camera studies. [3,4,5] However, since depth cameras can detect the planar surface of the checkerboard and not the black and white pattern on the checkerboard, in some studies infrared images of the depth camera are used instead of the depth images to calculate the parameters of the camera.[6,7,8] In some studies, depth images are used directly with the use of checkerboard like 3-dimensional special mechanisms to find the calibration parameters of depth cameras. [9]

Since in the use of a 2-dimensional planar calibration object such as a checkerboard, the optical center (o_x, o_y) cannot be calculated with sufficient precision [10] as explained in the study of Remondino and Fraser, and therefore, additional correction steps are needed after transferring the point clouds to the world coordinates [11], in this study, a 3-dimensional tetrahedron calibration object containing 4 equiradius spheres at its corners was designed. The projection matrix method was rearranged using the depth information given by the depth camera and the required minimum number of matching points was reduced from 6 to 4. Parameters were calculated for 2 different positions of the calibration object and the depth images were converted to camera and world coordinates. In order to reduce the matching errors observed in the world coordinates, the extrinsic parameters were recalculated using the camera coordinates obtained with the average intrinsic parameters and world coordinates. In experimental studies using only four matching points, the point clouds of the cameras were successfully matched without the need for any additional correction. To the best of our knowledge, the number of matching points required to achieve successful 3D reconstruction is less in our study compared to similar studies, since we used depth values.

In Section II, the designed tetrahedron calibration object was explained, in Section III the steps of the developed method were specified, in Section IV the experimental results were given and in the last section concluding remarks were made.

II. Tetrahedron Calibration Object

A calibration object (Fig. 1) was designed to find the calibration parameters of a depth camera by using only depth values. This object consists of 4 spheres (red (Rc), green (Gc), blue (Bc) and white (Wc)) with a radius of 73.2 mm and 50 cm long sticks located between the spheres.

The white arrows in Fig. 1 show the world coordinates, the origin point was chosen as the midpoint of the centers of the red, green and blue spheres. The direction of the +X axis was chosen from origin point to the center point of the red sphere and the direction of the +Y axis was chosen from origin point to the center point of the white sphere. So, the Z axis of the world coordinates became parallel to the stick between the green and the blue spheres. The center points of the spheres

978-1-6654-8894-5/22/$31.00 ©2022 IEEE

were calculated in world coordinates by using the known lengths and the angle between the neighboring sticks being 60 degrees.

The calculated coordinates of the centers of the spheres in cm are as following.

$$Rc = (37.3212 , 0 , 0)$$

$$Gc = (-18.6606 , 0 , -32.3212)$$

$$Bc = (-18.6606 , 0 , 32.3212)$$

$$Wc = (0 , 52.7975 , 0)$$

III. Calibration Procedure

Steps 1-5 of the following procedure was developed to find the coordinates of the sphere centers in the image coordinates and step 6 was developed to calculate the calibration parameters of the depth camera by using the known depth information.

Step 1: (For each camera) Using 5 depth frames of the same scene, finding the average of them and using it to eliminate the possible points that cannot be measured by Kinect in some frames.

Step 2: Manually selecting the left-upper and right-bottom of each sphere. (Fig. 2)

The colors of spheres in averaged depth images were assumed to be known. Each selected sphere was used as separate images after applying averaging (window size is 7x7) and median filters (window size is 3x3).

Step 3: Finding the positions of the smallest valued pixel/s (corresponding to the pixel/s closest to the camera since the depth image was used) in each determined sphere image and calculating the averages of the positions to limit the surface fitting step.

Step 4: Removing the unnecessary pixels whose value is greater than the determined minimum pixel value + sphere radius before the surface fitting step.

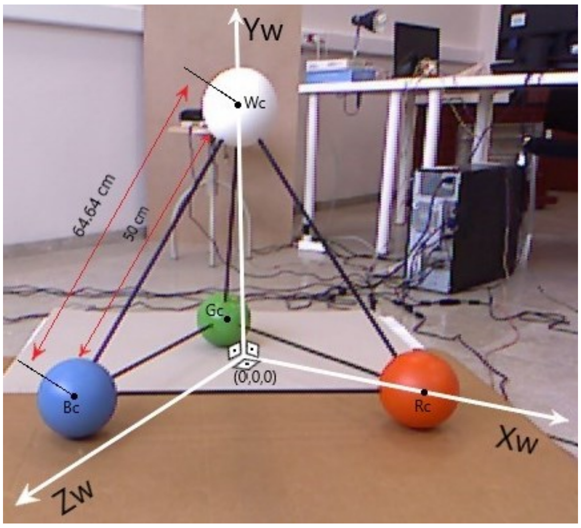


Fig. 1. Tetrahedron calibration object

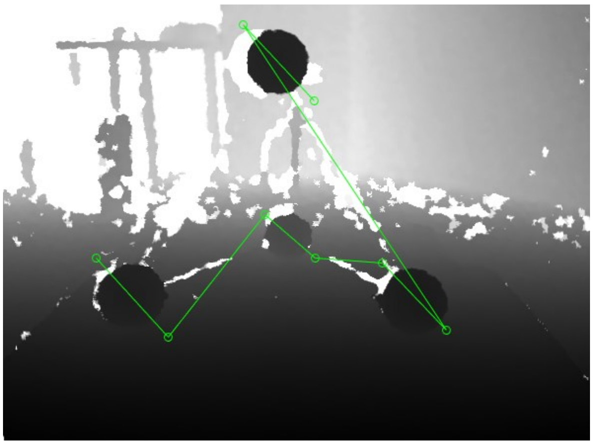


Fig. 2. Step 2 of the calibration procedure

The goal is to accurately detect spheres and their center points, so it is very important to delete unnecessary background pixels that do not belong to the sphere before the sphere detection step.

Step 5: Applying surface fitting.

Surface fitting process can be applied to the depth images of the spheres to detect the coordinates of the minimum valued point of the sphere since they have the depth information as the pixel values.

It is not appropriate to use sphere fitting algorithms to detect spheres and their center coordinates in depth images since the calibration parameters of the depth camera (focal lengths) are not yet known and the pixels are not in 3D form. For this reason, surface fitting was applied to pixels that do not have the characteristics of a spherical object in the X and Y axes, but only have the spherical characteristics in the Z axis due to the depth information, acquired by sensor.

MATLAB's curve fitting toolbox [12] is used with 5x5 degree polynomial fitting options. The fitting type was chosen as a 5th degree polynomial and surface fitting was performed using the linear least squares method.

Top figure in Fig. 3 shows the raw pixel points of a sphere and the surface fitted to these points. The fitted surface was limited to a window according to the coordinates found in Step 3, and the coordinates of the minimum valued point inside this window were taken as the coordinate of the pixel that represents the closest point of the sphere to the camera.

Bottom figure in Fig. 3 shows an example of a detected pixel coordinates of the closest point to camera. The radius of the sphere is known, and after calculating the pixel coordinate of the closest point, the center of the sphere can be found as

Xc =X coordinate of the closest point

Yc = Y coordinate of the closest point

Zc = Radius of the sphere + Z coordinate of the closest point (depth value)

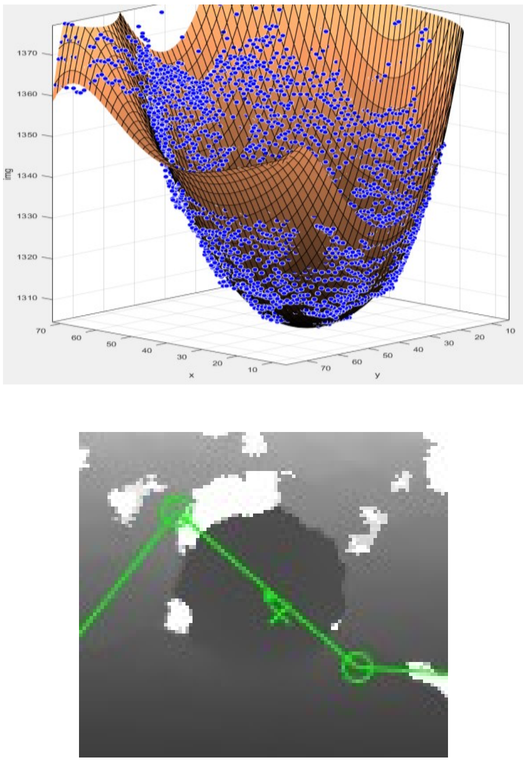


Fig. 3. (Top) Raw pixel (depth) values of an example sphere and fitted surface. (Bottom) Detected pixel point closest to camera ("**x**" symbol is from mean of the coordinates of the minimum valued pixels, "**.**" found from the surface fitting)

The closest points coordinates obtained by surface fitting were preferred because they gave more accurate results after matching the point clouds compared to the coordinates calculated in Step 3.

Step 6: Using the coordinates found in step 5 and the known world coordinates of the sphere centers to calculate the depth camera parameters.

The projection matrix method was used to calculate the calibration parameters, different from the original version, as described below.

The projection matrix method [13] includes estimating the 3x4 sized projection matrix M connecting the world (X_i^w, Y_i^w, Z_i^w) and image (x , y) coordinates and calculating the calibration parameters as closed form functions of the elements of the matrix.

The focal length scaled by horizontal and vertical pixel size of the camera are f_x and f_y, the optical center coordinates are o_x and o_y, the elements of the rotation matrix are r_{ij} and the elements of the translation vector are T_x, T_y and T_z, which make up the projection matrix M.

The equations are written as following.

$$x = \frac{u_i}{w_i} \quad and \quad y = \frac{v_i}{w_i} \qquad (1)$$

$$\begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix} = M_{int} M_{ext} \begin{pmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \end{pmatrix} = M \begin{pmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \end{pmatrix} \qquad (2)$$

M_{int} and M_{ext} are matrices which form the following projection matrix M, containing intrinsic and extrinsic calibration parameters, respectively.

$$M = \begin{bmatrix} -f_x r_{11} + o_x r_{31} & -f_x r_{12} + o_x r_{32} & -f_x r_{13} + o_x r_{33} & -f_x T_x + o_x T_z \\ -f_y r_{21} + o_y r_{31} & -f_y r_{22} + o_y r_{32} & -f_y r_{23} + o_y r_{33} & -f_y T_y + o_y T_z \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix}$$

The extrinsic parameters matrix M_{ext} connects a point P in the camera coordinates X_i^c, Y_i^c, $Z_i^c$and the world coordinates by using the (3).

$$P_c(X_i^c, Y_i^c, Z_i^c) = M_{ext} P_w(X_i^w, Y_i^w, Z_i^w) \qquad (3)$$

$$\begin{bmatrix} X_i^c \\ Y_i^c \\ Z_i^c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \end{bmatrix}$$

In this study, depth images which contain only depth information were used to calculate the calibration parameters of depth cameras so, the Z_i^c values of the sphere centers were known.

$$Z_i^c = X_i^w r_{31} + Y_i^w r_{32} + Z_i^w r_{33} + T_z \qquad (4)$$

There are 4 unknown parameters in the equation (4) and by using 4 matching points between the world and image coordinates, the r_{31}, r_{32}, r_{33} and $T_z$elements, which form the bottom row of the M projection matrix, can be calculated by solving the following.

$$Ax = b \qquad x = A^{-1} b \qquad (5)$$

$$A = \begin{bmatrix} X_1^w & Y_1^w & Z_1^w & 1 \\ X_2^w & Y_2^w & Z_2^w & 1 \\ \vdots & & & \vdots \end{bmatrix}, \quad b = \begin{bmatrix} Z_1^c \\ Z_2^c \\ \vdots \end{bmatrix}, \quad x = \begin{bmatrix} r_{31} \\ r_{32} \\ r_{33} \\ T_z \end{bmatrix}$$

Since 4 of the unknown elements of the M projection matrix were calculated using the depth information, there were 8 matrix elements remained to be calculated. The minimum number of matching points required between world and image coordinates to calculate the remaining 8 unknown elements became 4.

While the minimum required number of matching points was 6 in the original projection matrix method, in our study this number was reduced to 4 thanks to the known depth values. Thus, the tetrahedron calibration object with 4 equiradius spheres we created can be used with a single exposure to calculate the calibration parameters of any depth camera.

For the calculation of the calibration parameters, the remaining elements of the M projection m_{ij}, $i = 1{,}2$ and $j = 1{,}2{,}3{,}4$ must be calculated first. Using the world and image coordinates of the sphere centers, the unknown elements can be calculated by using (5) and the following.

$$A = \begin{bmatrix} X_1^w & Y_1^w & Z_1^w & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_1^w & Y_1^w & Z_1^w & 1 \\ X_2^w & Y_2^w & Z_2^w & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_2^w & Y_2^w & Z_2^w & 1 \\ \vdots & & & \vdots & & \vdots & & \vdots \end{bmatrix}, x = \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{14} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{24} \end{bmatrix}$$

$$b = \begin{bmatrix} m_{31}x_1X_1^w + m_{32}x_1Y_1^w + m_{33}x_1Z_1^w + m_{34}x_1 \\ m_{31}y_1X_1^w + m_{32}y_1Y_1^w + m_{33}y_1Z_1^w + m_{34}y_1 \\ m_{31}x_2X_2^w + m_{32}x_2Y_2^w + m_{33}x_2Z_2^w + m_{34}x_2 \\ m_{31}y_2X_2^w + m_{32}y_2Y_2^w + m_{33}y_2Z_2^w + m_{34}y_2 \\ \vdots \end{bmatrix}$$

The scale factor is defined as

$$|\gamma| = \sqrt{m_{31}^2 + m_{32}^2 + m_{33}^2} = |\gamma|\sqrt{r_{31}^2 + r_{32}^2 + r_{33}^2}$$

and the elements of the matrix M is divided by $|\gamma|$. By defining the updated elements of the M matrix as

$$\boldsymbol{a}_1 = [m_{11}, m_{12}, m_{13}]^T$$

$$\boldsymbol{a}_2 = [m_{21}, m_{22}, m_{23}]^T$$

$$\boldsymbol{a}_3 = [m_{31}, m_{32}, m_{33}]^T$$

$$\boldsymbol{a}_4 = [m_{14}, m_{24}, m_{34}]^T,$$

calibration parameters of the depth camera can be calculated as following equations.

$$o_x = \boldsymbol{a}_1^T\boldsymbol{a}_3 \qquad o_y = \boldsymbol{a}_2^T\boldsymbol{a}_3 \tag{6}$$

$$f_x = \sqrt{\boldsymbol{a}_1^T\boldsymbol{a}_1 - o_x^2} \qquad f_y = \sqrt{\boldsymbol{a}_2^T\boldsymbol{a}_2 - o_y^2} \tag{7}$$

$$\begin{gathered} r_{1i} = (o_x m_{3i} - m_{1i})/f_x \\ r_{2i} = (o_y m_{3i} - m_{2i})/f_y \qquad i = 1{,}2{,}3 \\ r_{3i} = m_{3i} \end{gathered} \tag{8}$$

$$\begin{gathered} T_z = m_{34} \\ T_x = (o_x T_z - m_{14})/f_x \\ T_y = (o_y T_z - m_{24})/f_y \end{gathered} \tag{9}$$

IV. Experimental Works and Results

Depth images were taken from the tetrahedron calibration object at 2 (to test the results with the minimum number of positions) different positions and the average of the results obtained from these 2 positions was used to calculate the intrinsic parameters. The reason for using the mean value here was to minimize the errors that may result from misdetection of the closest point of the sphere due to possible noise in the depth images.

Average intrinsic parameters were calculated with the formulas given in section III, then extrinsic parameters were calculated, and depth images were transformed from image coordinates to world coordinates. The calculated intrinsic parameters are as following.

$$o_x = [313.05, 323.33], \qquad o_y = [249.79, 239.03]$$

$$o_{x_{ort}} = 318.19\,pixel, \qquad o_{y_{ort}} = 244.41\,pixel$$

$$f_x = [613.15, 615.45], \qquad f_y = [600.73{,}626.80]$$

$$f_{x_{ort}} = 614.30\,pixel, \qquad f_{y_{ort}} = 613.76\,pixel$$

Fig. 4 shows the point clouds of the depth images in the world coordinates. To distinguish easily, point clouds were made in yellow, pink, blue and white colors for images taken from cameras 1, 2, 3 and 4, respectively.

As can be seen from the point clouds in Figure 4, there were some matching issues between the cameras. In order to find the parameters (intrinsic and/or extrinsic) that cause this problem, first the sphere images were transformed into camera coordinates with the intrinsic parameters, and then the radii in the obtained point clouds were calculated by the sphere fitting method. [14,15] The radius range of the detected spheres was found to be 72 mm – 74 mm. Also, the distance between the centers of the detected spheres for the point clouds of all cameras was checked. Since the radius and distance values in the point clouds were consistent with the tetrahedron calibration object, it was understood that the calculated average intrinsic parameters were found correctly and the matching problem in the world coordinates was caused by the calculation of the extrinsic parameters (rotation and translation matrices).

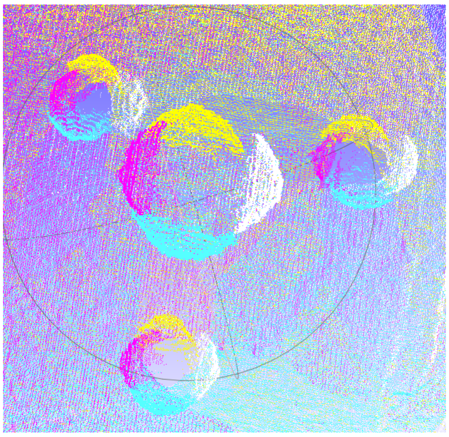


Fig. 4. Point clouds in world coordinates

Another step to find more accurate rotation and translation matrices was developed by using the image coordinates and the world coordinates of the center point of the spheres. Only the coordinates of the sphere centers detected in the depth images were transformed into the camera coordinates (X^c, Y^c, Z^c) by using the calculated mean intrinsic parameters.

$$Z^c = depth\ of\ the\ detected\ pixel + \ sphere\ radius$$

$$X^c = {(o_x - x)Z^c}\big/{f_x} \qquad Y^c = {(o_y - y)Z^c}\big/{f_y} \qquad (10)$$

Camera coordinates of the center of the red sphere is (X_R^c, Y_R^c, Z_R^c), the green sphere is (X_G^c, Y_G^c, Z_G^c), the blue sphere is (X_B^c, Y_B^c, Z_B^c) and the white sphere is (X_w^c, Y_w^c, Z_w^c). The world coordinates of the center of the red sphere is (X_R^w, Y_R^w, Z_R^w), the green sphere is (X_G^w, Y_G^w, Z_G^w), the blue sphere is (X_B^w, Y_B^w, Z_B^w) and the white sphere is (X_w^w, Y_w^w, Z_w^w). Transformation from the camera to the world coordinates is

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{bmatrix}$$

The last row elements of the extrinsic matrix are calculated before by using the depth values of the sphere centers. The remaining unknown eight parameters can be calculated by using Ax=b where

$$A = \begin{bmatrix} X_R^w & Y_R^w & Z_R^w & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_R^w & Y_R^w & Z_R^w & 1 \\ X_G^w & Y_G^w & Z_G^w & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X_G^w & Y_G^w & Z_G^w & 1 \\ \vdots & & & \vdots & & \vdots & & \vdots \end{bmatrix}$$

$$x = [r_{11} \quad r_{12} \quad r_{13} \quad T_x \quad r_{21} \quad r_{22} \quad r_{23} \quad T_y]^T$$

$$b = [X_R^c \quad Y_R^c \quad X_G^c \quad Y_G^c \quad X_B^c \quad Y_B^c \quad X_w^c \quad Y_w^c]^T$$

By using the calculated intrinsic parameters, updated extrinsic parameters and 4 matching points, the depth images were transformed into the world coordinates. The point clouds of different cameras in Fig. 5 show that the matching errors are largely eliminated compared to the results in Fig. 4, and it is understood that the recalculated extrinsic parameters are obtained correctly.

In our previous study [8], the infrared images of the planar checkerboard (Fig. 6 (a)) mentioned in the introduction part were used to calculate the calibration parameters of the depth camera. The matching results shown in the Fig. 6 (b) was obtained only with the calculated parameters using the Bouguet's toolbox [2], without any extra processing. Iterative Closest Points (ICP) method, which is one of the most commonly used methods to match the point clouds, was used to reduce the matching error between point clouds and only then sufficient matching results were obtained. (Fig. 6 (c))

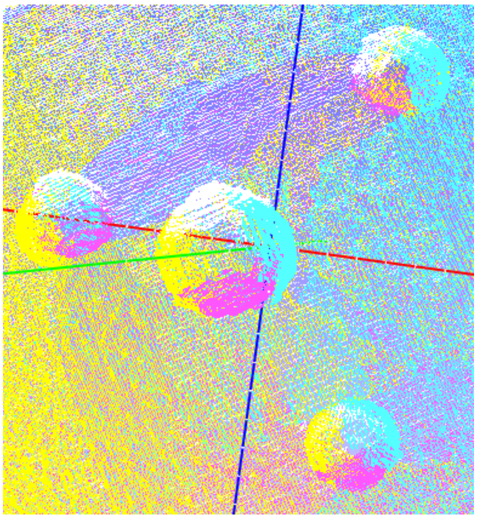
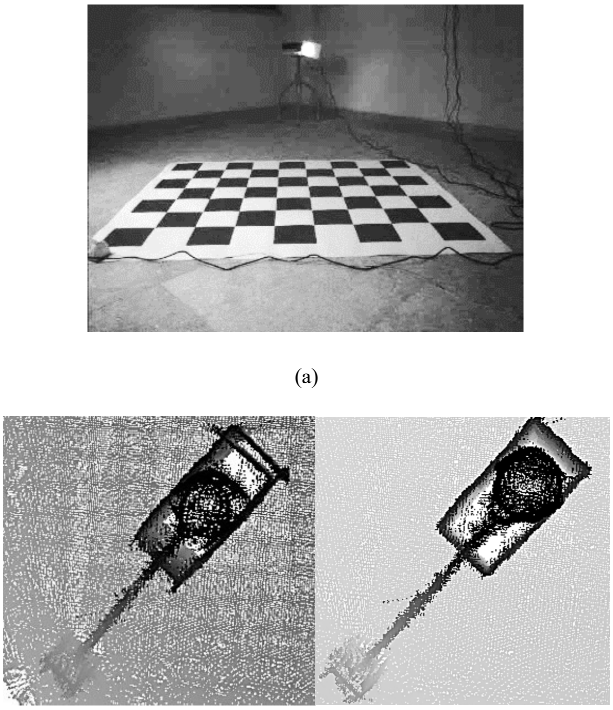


Fig. 5. Point clouds in world coordinates (obtained by using the updated extrinsics parameters)



(a)

(b)

(c)

Fig. 6. (a) Infrared checkerboard (b)Point clouds in world coordinates (obtained by only applying the parameters) (c) After applying ICP method [8]

Table 1 shows the intrinsic parameters calculated with planar checkerboard and tetrahedron calibration object approaches. There are differences between the two approaches, especially in terms of the calculated focal lengths. This is because when using an IR image, only the parameters of the IR camera are calculated, not the depth camera. Actually, the device uses an IR camera for depth values, but it also uses different additional operations to calculate depth values such as finding disparities, etc. In order to use the pinhole camera model successfully, the depth images should

TABLE I. INTRINSIC PARAMETERS OBTAINED WITH PLANAR CHECKERBOARD AND TETRAHEDRON CALIBRATION OBJECT

	Planar object (Infrared Image)[8]	Tetrahedron object (Depth image)
fx	586.4 pixels	614.30 pixels
fy	587.5 pixels	613.76 pixels
ox	320.6 pixels	318.19 pixels
oy	246.5 pixels	244.41 pixels

be considered as if they were taken with another camera independent of the IR camera and the parameters should be calculated accordingly. In this context, this novel approach, using only the depth images of the tetrahedron calibration object described in this paper, can be applied directly to the pinhole model without the need for additional methods used to reduce matching errors.

V. CONCLUSION

In our study, a 3-dimensional tetrahedron calibration object, containing four equiradius spheres, was designed in order to eliminate the matching errors occur when the depth camera is calibrated by using infrared images of a planar checkerboard. Spheres in the depth images were selected and the average of the coordinates of the minimum valued pixels was calculated to limit the surface fitting step. The surface fitting process was applied to the selected spheres using the curve fitting function of MATLAB. The coordinate of the minimum valued point of the fitted surface was chosen as the closest point of the sphere to the camera. The depth value of the sphere center was taken as the sum of the pixel value of the closest point and the radius of the sphere.

Using the depth value measurements, the lowest row elements of the projection matrix M of the projection matrix method were easily calculated with four matching points between the world and the camera coordinates (Z axis). With the use of depth information, the minimum number of matching points required by the original projection matrix method was reduced from 6 to 4, making the calibration process much faster and easier to apply.

Intrinsic and extrinsic parameters were calculated using the row elements of the M matrix. The calibration parameters found were applied to the depth images, but it was observed that the point clouds of the different cameras obtained did not match sufficiently. In order to find out which parameters caused the matching problem; the sphere fitting process was applied to the spheres in the camera coordinates and results consistent with the calibration object were obtained. Thus, it was understood that the problem was not in the intrinsic parameters, but in the extrinsic parameters. The extrinsic parameters matrix was recalculated and updated using only the world coordinates of the sphere centers and the calculated camera coordinates. After applying the updated parameters to the point clouds in the camera coordinates the matching error was decreased.

A novel calibration object and method has been introduced and based on the successful matching results obtained with the updated parameters. This approach can be used in all depth cameras and can provide successful results even with very few matching points and positions.

REFERENCES

[1] Z. Zhang, "A Flexible New Technique for Camera Calibration", IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), pp. 1330–1334, 2000.

[2] J. Y. Bouguet, "Camera Calibration Toolbox for MATLAB", accessed 09 February 2022, http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#parameters

[3] E. Auvinet, J. Meunier and F. Multon, "Multiple depth cameras calibration and body volume reconstruction for gait analysis.", 10.1109/ISSPA.2012.6310598, pp. 478-483. 2012.

[4] N. Burrus, "Kinect Calibration", accessed 09 February 2022, http://nicolas.burrus.name/index.php/Research/KinectCalibration

[5] R. Avetisyan, M. Willert, S. Ohl and O. Staadt, "Calibration of Depth Camera Arrays." 10.13140/2.1.2488.6720, 2014.

[6] J. Smisek, M. Jancosek, and T. Pajdla, "3D with Kinect." 10.1007/978-1-4471-4640-7_1, 2013.

[7] Reimann, A., "Intrinsic calibration of the Kinect cameras", accessed 09 February 2022, http://wiki.ros.org/action/show/openni_launch/Tutorials/IntrinsicCalibration?action=show&redirect=openni_camera%2Fcalibration

[8] E. Tunçer and Ş. Gümüştekin, "3D Object Reconstruction Using Sequentially Activated Multiple Depth Cameras", 28th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, doi: 10.1109/SIU49456.2020.9302191, 2020

[9] C. Zhang, T. Huang, and Q. Zhao, "A New Model of RGB-D Camera Calibration Based On 3D Control Field", Sensors, 19. 5082. 10.3390/s19235082, 2019.

[10] F. Remondino and C. Fraser, "Digital Camera Calibration Methods: Considerations And Comparisons", ISPRS Commission V Symposium 'Image Engineering and Vision Metrology',2006.

[11] P. Besl and H.D. McKay, "A method for registration of 3-D shapes.", IEEE Trans Pattern Anal Mach Intell. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 14., 10.1109/34.121791, pp. 239-256, 1992.

[12] Curve Fitting Toolbox, accessed 09 February 2022, https://www.mathworks.com/help/curvefit/index.html?s_tid=CRUX_lftnav

[13] E. Trucco, and A. Verri, "Introductory techniques for 3-D computer vision", *Prentice Hall*, ISBN: 978-0-13-261108-4, 1998.

[14] The Computational Geometry Algorithms Library, accessed 20 December 2020, https://www.cgal.org/

[15] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection." Comput. Graph. Forum. 26., 10.1111/j.1467-8659.2007.01016.x., pp. 214-226, 2007.