

Article

Reduced Calibration Strategy Using a Basketball for RGB-D Cameras

Luis-Rogelio Roman-Rivera ^{*,†} , Israel Sotelo-Rodríguez [†], Jesus Carlos Pedraza-Ortega [†] ,
Marco Antonio Aceves-Fernandez [†] , Juan Manuel Ramos-Arreguín [†]  and Efrén Gorrostieta-Hurtado [†] 

Facultad de Ingeniería, Universidad Autónoma de Querétaro, Cerro de las Campanas S/N, Querétaro C.P. 76010, Mexico; isotelo17@alumnos.uaq.mx (I.S.-R.); carlos.pedraza@uaq.mx (J.C.P.-O.); marco.aceves@uaq.mx (M.A.A.-F.); juan.ramos@uaq.mx (J.M.R.-A.); efrén.gorrostieta@uaq.edu.mx (E.-G.H.)

* Correspondence: lroman26@alumnos.uaq.mx

† These authors contributed equally to this work.

Abstract: RGB-D cameras produce depth and color information commonly used in the 3D reconstruction and vision computer areas. Different cameras with the same model usually produce images with different calibration errors. The color and depth layer usually requires calibration to minimize alignment errors, adjust precision, and improve data quality in general. Standard calibration protocols for RGB-D cameras require a controlled environment to allow operators to take many RGB and depth pair images as an input for calibration frameworks making the calibration protocol challenging to implement without ideal conditions and the operator experience. In this work, we proposed a novel strategy that simplifies the calibration protocol by requiring fewer images than other methods. Our strategy uses an ordinary object, a know-size basketball, as a ground truth sphere geometry during the calibration. Our experiments show comparable results requiring fewer images and non-ideal scene conditions than a reference method to align color and depth image layers.

Keywords: RGB-D camera; RGB-D camera calibration; spherical object; 3D reconstruction; sphere detection

MSC: 65D19; 53C38



Citation: Roman-Rivera, L.-R.; Sotelo-Rodríguez, I.; Pedraza-Ortega, J.C.; Aceves-Fernandez, M.A.; Ramos-Arreguín, J.M.; Gorrostieta-Hurtado, E. Reduced Calibration Strategy Using a Basketball for RGB-D Cameras. *Mathematics* **2022**, *10*, 2085. <https://doi.org/10.3390/math10122085>

Academic Editors: Xiangtao Zheng, Jinchang Ren and Ling Wang

Received: 19 May 2022

Accepted: 13 June 2022

Published: 16 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

RGB-D cameras are becoming popular due to their availability, size, and accessible cost; there are different choices in the market from different brands. Microsoft popularized this type of camera with the Kinect [1] camera used initially for gaming (Figure 1a) in 2010, and its 3D depth technology used structured light [2]. It had a visual range from 0.8 to 4 m, producing 640×480 depth images. However, scientific research was later possible thanks to a personal computer's Kinect software development kit allowing this camera to be connected directly to a personal computer. Data produced by the Kinect could finally be processed for scientific purposes [3]. Two years later, Kinect V2 replaced Kinect V1 for the Xbox One gaming console with a range from 0.5 to 4.5 meters, a resolution of 512×424 , and a 1080p RGB camera [3,4].

After Kinect, other RGB-D alternatives appeared. Intel released its Realsense camera line [3,5] reducing camera size and power requirements [4], but providing a software development kit to work with the data directly from a personal computer, allowing production of 3D cloud points quickly (Figure 1b). Intel Realsense D415 RGB-D camera has a visual range from 0.16 to 10 m and produces depth images with a resolution of 1280×720 . Realsense D435 has a visual range from 0.2 to 4.5 m. In these cameras, depth computation is performed by an integrated ASIC (Application-Specific Integrated Circuit) [4].

Conversely, Occipital offered its structure camera [6,7] (Figure 1c), Structure core camera from Occipital has a visual range from 0.3 to 5 m, and it produces depth images

with a resolution of 1280×960 [6], following the same philosophy as Intel, but available mainly in Apple's ecosystem.

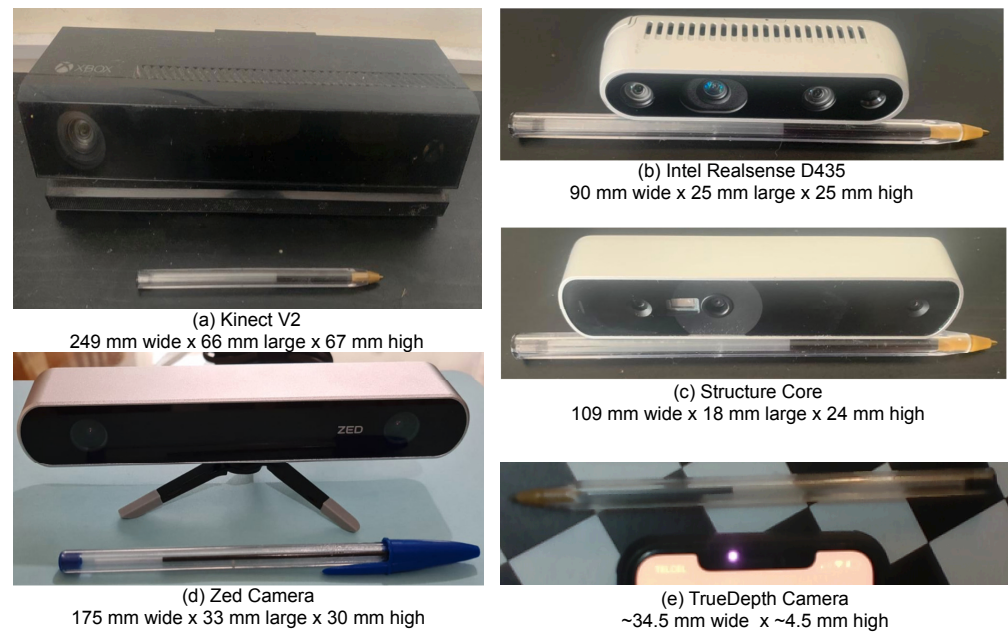


Figure 1. RBB-D cameras, (a) Kinect, (b) Realsense D435, (c) StructureCore, (d) Zed Camera, (e) TrueDepth Camera.

Stereolab offered ZED RGBD cameras using a different approach, generating its depth layer from the pixels disparities from two RGB images (Figure 1d). Stereolab's original ZED camera has a visual range from 0.5 to 20 m, and it produces depth images of a resolution of 4416×1242 . In the figure, a ZED 2i camera is shown, it has an accelerometer, gyroscope, barometer, magnetometer, and temperature sensors included [8].

In 2018, Apple equipped one of its mobile phones with the TrueDepth (Figure 1e) camera, the main purpose of this camera was for authentication from the 3D geometry and textures data [9].

RGB-D cameras produce two different layers of information, a color layer (one or two RGB images) and a depth layer (generally represented as one image containing depth information) (Figure 2).

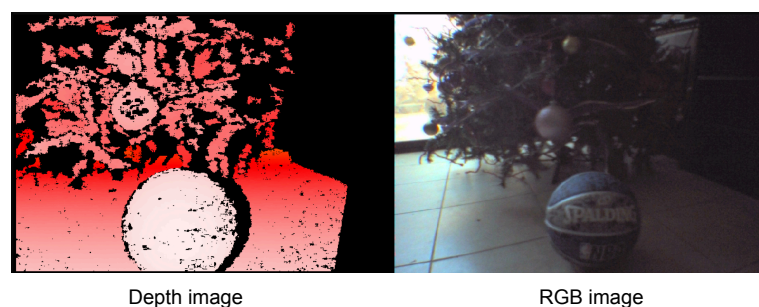


Figure 2. In the figure at the left, there is an example of a depth image representation in red tones, closer objects appear in light red, and farther objects are in darker red. The RGB camera component takes the corresponding color image of the same scene shown on the right side.

Both layers, color, and depth correspond in an overlapping area given rotation and translation matrices parameters. A factory camera calibration process generally provides these parameters as default values with brand new cameras. However, in some applications, factory parameters are not good enough. Furthermore, having different RGB-D

cameras of the exact brand and models show different calibration errors of the same scene captured in their images with different precision and exactitude values. Agriculture [10], 3D reconstruction [11–16], 3D navigation [17–20], robotics [21], augmented reality (AR) [22–24], autonomous driving [25,26], object recognition [27–29], computer vision [30], are some research areas using data from these cameras. Considering the RGB-D camera’s available features and limitations, minimizing calibration error is an essential step for more delicate applications.

In the RGB color layer, a geometric error is produced by lens manufacturing imperfections causing the barrel or pincushion distortion showing curves instead of perfect lines. This kind of distortion affects the geometry of the objects in the scene [31] (Figure 3).

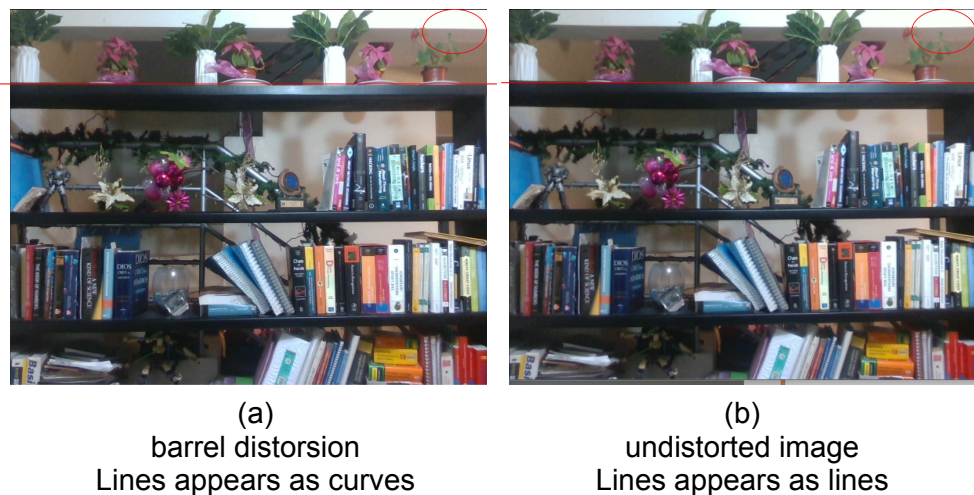
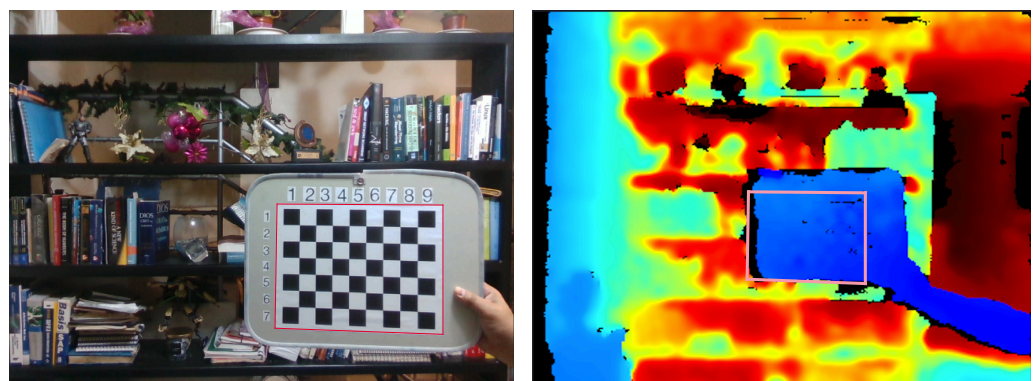


Figure 3. Barrel distortion appears at the left of the image. The main characteristic of this type of distortion is straight lines appearing as curves with the outer side from the center to the edges of the image (a). On the right side (b), the same scene image after an undistortion correction [31].

When aligning both layers to correspond to textures from the color layer with 3D objects in the depth layer, some areas and/or objects do not show only their textures, but appear misaligned, mixing other texture regions from other objects in the scene [31] (Figure 4).



Alignment between color and depth images.

Figure 4. An alignment error example is visible in the above figure. At the left, there is a chessboard which perimeter corresponds to the red square at the right when overlapping textures in the depth image. In the right, in the depth data, chessboard corners do not correspond to the chessboard [31].

A simplified calibration protocol should provide a way to keep the RGB-D cameras inside the precision and exactitude desired for specific applications in non-controlled scenes. In state-of-the-art, there are multiple calibration protocols with novel methods

that can adjust parameter values to finer precisions and exactitudes other than the factory values. Nevertheless, all of them require a controlled scene, more delicate instruments, a calibration protocol operator expert, and a considerable amount of samples during the calibration protocol [32]; consequently, the calibration process is not easy to follow for a non-expert operator.

The RGB-D camera calibration method proposed by [33] requires between twenty and sixty color-depth pairs. The proposed methodology uses a checkerboard as a ground-truth pattern in the color and depth layers. The strategy created by [34] uses more than one hundred color-depth pairs, and their calibration protocol requires sampling images with distance differences from the camera in a progressive way.

Recently, some RGB-D camera calibration methods are innovating using a sphere as geometry to be detected in the color and depth layers, but still requiring a considerable amount of pair samples during the calibration protocol, ref. [35] Figure 5 requires 130 color-depth pairs. Furthermore, some posterior phases require images to be manually selected and masked so the calibration toolboxes can process the data accordingly.

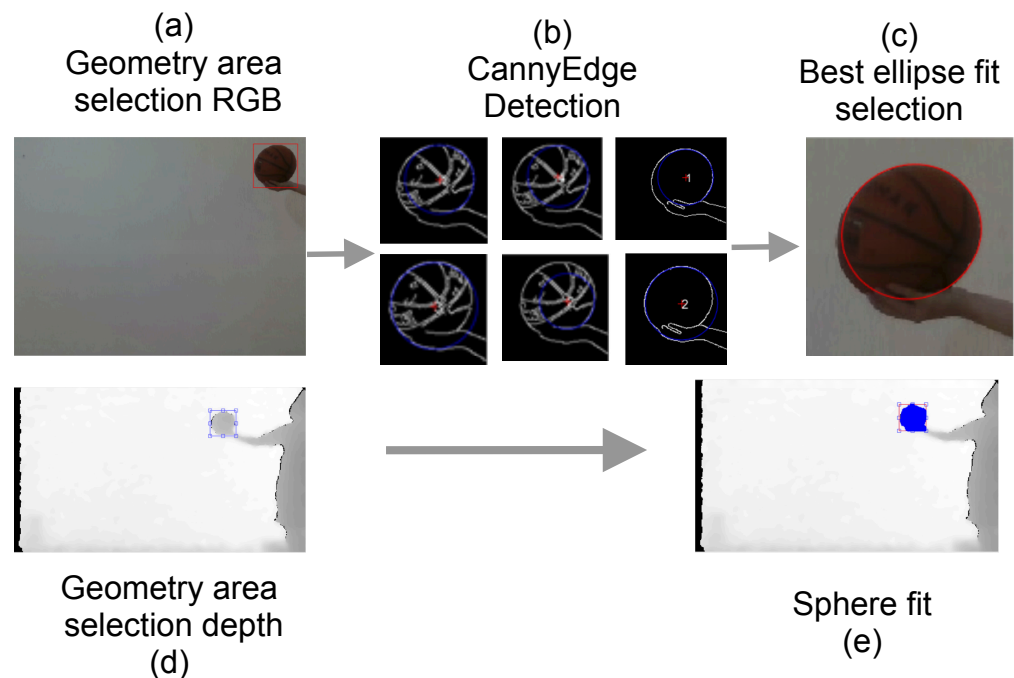


Figure 5. Staranowicz's method detects a basketball as a geometry object in both layers, color, and depth, but it requires one hundred and thirty image pairs during the calibration protocol. This requirement increases the calibration protocol complexity [35,36].

Different proposals in state of the art as [37] estimate relative poses of multiple RGB-D cameras nevertheless, it uses 2D keypoints, descriptor-based patterns, and locates 3D depth matches solving an optimization model comparing images from a group of different RGB-D cameras, aligning information between them. The calibration method of [38] calibrates an RGB-D constellation using static markers to align data produced by each camera in the constellation using a homography and an ICP variant. The work by [39] improves depth quality using a novel method to match infrared images and [40] focuses on calibrating depth layer utilizing a checkerboard in the infrared images to improve depth precision.

This paper follows a similar approach to [35,36] using a sphere to find a scene's geometry pattern in information layers, depth, and color. We use a basketball of known size as a sphere geometry, QUEsT method [41] has been adapted in a novel way to use a combination of pair images with depth and color information as an input, combining novel methods in the calibration process to correspond a minimum number of ellipse's centers in the color layer to sphere's centers in both layers, simplifying the calibration

protocol to get the rotation and translation matrices. We compare our method with another sphere calibration algorithm showing advantages in the calibration protocol steps in a non-controlled scene.

Our strategy simplifies calibration as follows:

- It requires a minimal amount of images.
- An ordinary object is used as a pattern (a basketball of known size).
- A sphere can be positioned in the scene easily as the geometry because it is equally seen from any angle.
- It allows non-ideal conditions in the scene as natural illumination and non-uniform textures.

2. Materials and Methods

2.1. Hardware and Software

This work used a personal computer with Ubuntu 18 LTS, an AMD Ryzen 5600x CPU, 6 Cores, 12 threads, 3.7GHz 32MB L3 Cache 3MB L2 Cache, 16 GB RAM, NVIDIA GeForce RTX 3060 TI, 8GB GDDR6. 4864 CUDA Cores, Ubuntu 20.04.2 LTS OS, Docker 19.03.8, Python 3.9, furthermore a Macbook air laptop (Retina 13-inch, 2020) with a CPU 1.1 GHz Quad-Core Intel Core i5 and 8GB of RAM and Integrated graphics Intel Iris Plus Graphics 1536 MB, Mac OS Big Sur 10.13 Beta OS, Python 2.9 and Matlab 2018a and C++. A raspberry pi 4 with raspbian and realsense viewer to capture frames from the scene.

2.2. Rgb-D Camera and Scene

During all experiments, we use an Intel Realsense D435 with the specifications in Table 1:

Table 1. Intel Realsense specs [5].

| Feature | Description |
|------------------------|-----------------------|
| System interface type | USB C type C |
| Dimensions | 90 mm × 25 mm × 25 mm |
| Depth resolution | 1280 × 720 |
| Visual Range (min–max) | ~0.11 m–10 m |

Factory intrinsic parameters are in the camera configuration; distortion parameters are 0 by default, and images can vary in size depending on the type of USB connection used. Images created using the raspberry pi 4 have the following sizes: RGB 640 × 480, depth images: 424 × 240.

The camera calibration protocol requires setting up a tripod and an aluminum bar to stabilize and level the camera. Connect the USB C cable to the camera and adjust it to the correct position with a couple of nuts. Moreover, to settle the camera correctly, with the help of a bubble level, align horizontally using the aluminum bar, align vertically with the front of the camera. USB C cable connects the camera to the raspberry pi to see the scene and capture depth and color frame through realsense viewer GUI.

It is essential to note camera's setup must be performed in the final position before taking images samples from the scene, avoiding moving the camera as in Figure 6.

A basketball of a standard size of seven (circumference of 29.5 in) needs to be placed in the scene in different positions, and the basketball must be utterly present in each sample's images, color, and depth. Our comparison considers non-ideal conditions in the location, indirect solar illumination, non-uniform background textures, changes in global illumination due to natural weather and time differences between each sample, and a minimum amount of pair images has been a goal of the proposed method. Seventy pairs of depth and color images were taken from the scene, considering different basketball positions to compare our proposed method with [35,36], their Matlab toolbox has been used to process our own images and can be found in [42].

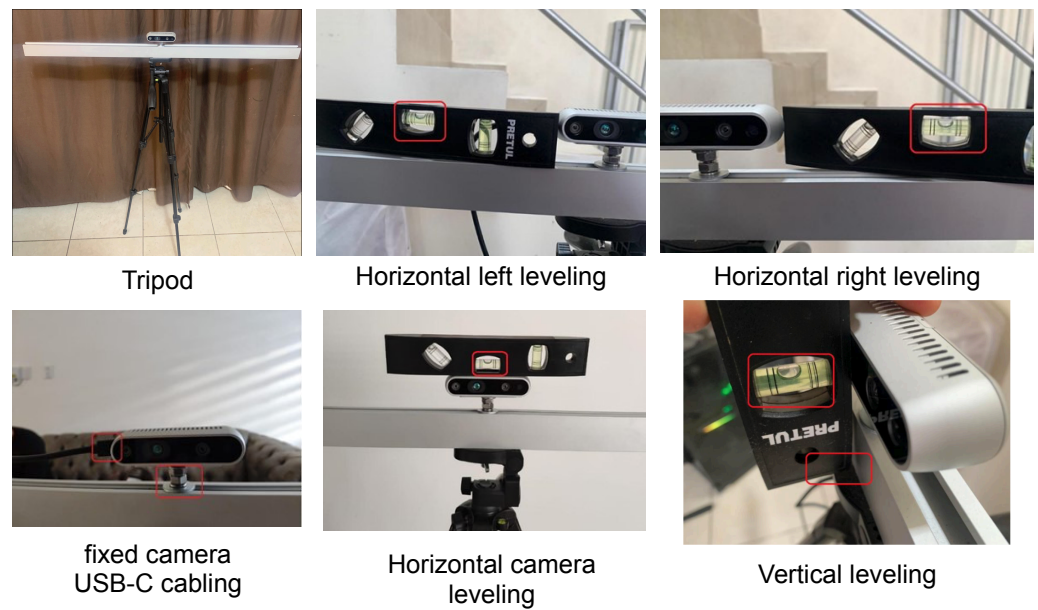


Figure 6. An example of adjusting and leveling the RGB-D camera.

2.3. Proposed Method

The proposed methodology requires undistorting the RGB image layer using the Zhang calibration strategy [43]. For this purpose, a set of at least 10 color images where a chessboard is visible needs to be sampled using RGB-D camera, as shown in Figure 7, we use a public toolbox available from [44] based in [45] to get a intrinsic matrix and undistortion coefficients.

The first phase of the proposed method in Figure 8 uses a well-established method [43] to de-distort color images and get an intrinsic matrix, in this step, ten different images showing a chessboard are enough.

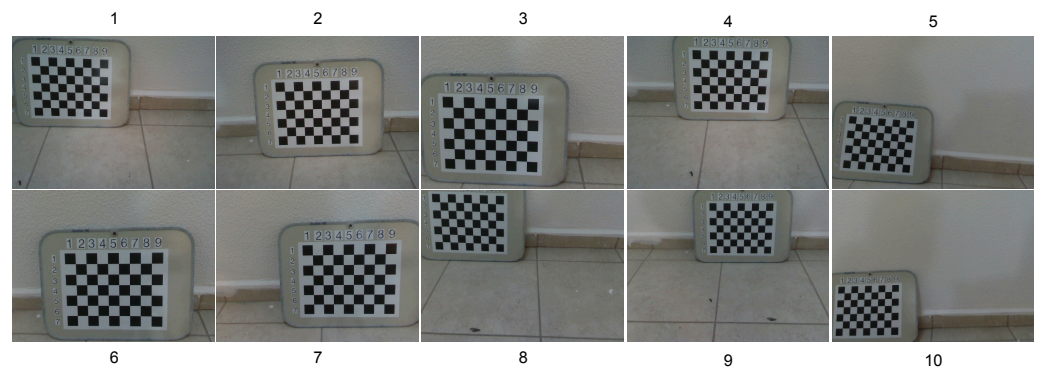


Figure 7. A set of ten different images with a chessboard in the scene are needed to use [43] in order to find the intrinsics and undistort RGB layer, implementation details is based in [45] and a toolkit is available by [44].

Furthermore, the process requires five pairs of depth and color images where a basketball is visible in the scene in each layer. Ellipses are found in the RGB images using Arc-support Line Segments Revisted [46], and we found spheres on depth images with our proposal for a known sphere size. Finally, our process uses Quaternion Based Camera Pose Estimation [41] to get rotation and translation matrices.

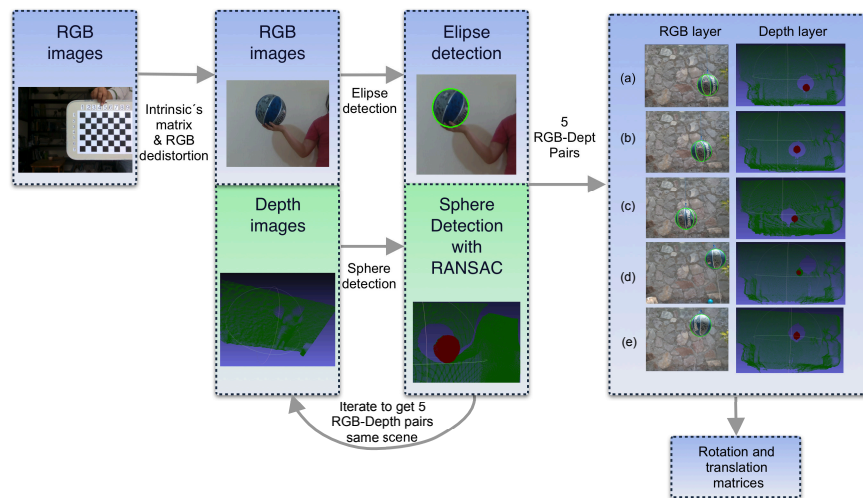


Figure 8. Proposed methodology. A minimum of five different pair images color-depth (a–e) are required to find ellipses and sphere centers.

The camera intrinsic matrix describes the relationship between world coordinates and image coordinates. The mathematical model uses five parameters focal length in the x direction as f_x and in the y direction as f_y , principal point or optical center in the x and y direction, the skew between the x and y axes. There are two general and equivalent forms of the intrinsic matrix, where pp_x and pp_y denote principal point coordinates, sometimes they can be found as c_x and c_y . A non-zero skew factor s implies that the x - and y -axis of the camera are not perpendicular to each other. We assume a skew factor of 0 for simplicity.

The intrinsic matrix denoted by \mathbf{K} is an upper-triangular matrix used to transform world coordinates to homogeneous image coordinates.

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Geometric error as seen on the left side of Figure 3 requires calibration using a distortion model to describe the deviations from a real camera that uses lenses and differs from the ideal pinhole camera. Undistortion coefficients [47] and intrinsics matrix are used to undistort an RGB image as show in Figure 3.

Radial distortion models convert between distorted and undistorted points and use a center of distortion (x_c, y_c) and a function $f(r)$:

$$x' = f(r) \cdot \cos(\theta) + x_c \tag{2}$$

$$y' = f(r) \cdot \sin(\theta) + y_c \tag{3}$$

The polynomial radial distortion model uses a polynomial:

$$f(r) = r \cdot \left(1 + p_1 \cdot r + p_2 \cdot r^2 + \dots + p_N \cdot r^N \right) = r \cdot \left(1 + \sum_{n=1}^N p_n \cdot r^n \right) \tag{4}$$

Further details of pin camera model and lens distortion correction can be found in [43,48].

The second phase of the proposed method in Figure 8 uses a novel approach to detect ellipses in the color images by [46]. Their approach allows detecting ellipses with better results than similar methods and simplifies the calibration protocol steps in this work. As shown in Figure 9, arc-support line segments revisited method [46] detect more than one ellipse in some cases. When more than one ellipse appears, an ellipse is selected considering

a correctness metric provided by their high-quality ellipse detection method as the best option, their public toolkit is available from [46].

Additionally, our adapted method to detect spheres in the depth layer uses a basketball of known size (standard size of seven about 12 cm of radius). It needs to be well inflated as required in the specifications to meet the legal size accordingly. These specifications can vary from each ball brand and model. Using the sphere equation with a radius r , the surface of which passes by three points p_1, p_2, p_3 , and the center is described by its coordinates $(a, b, c) = center$. Each point is describe by its own coordinates (x_i, y_i, z_i) as shown in Equation (5).

$$r^2 = (x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 \tag{5}$$

With Equation (5), $p_1 = (x_1, y_1, z_1)$, $p_2 = (x_2, y_2, z_2)$ and $p_3 = (x_3, y_3, z_3)$ as three different surface points, a known radius r of about 12 cm, we formulate an equation system to find the sphere $center = (a, b, c)$ as is shown in Equation (9).

$$\begin{aligned} r^2 &= (x_1 - a)^2 + (y_1 - b)^2 + (z_1 - c)^2 \\ r^2 &= (x_2 - a)^2 + (y_2 - b)^2 + (z_2 - c)^2 \\ r^2 &= (x_3 - a)^2 + (y_3 - b)^2 + (z_3 - c)^2 \end{aligned} \tag{6}$$

Five pairs of RGB (color) and depth raw images are showed at the left of the Figure 9, then these are processed to get ellipses from the color layer and spheres from the depth layer. As a final result of this phase, we get five correspondences from ellipse centers to spheres centers.

Developing Equation (9) as shown in (9), it is formulated in the form $Ax = b$ as shown in Equation (7) it can be used to find a, b, c .

$$\begin{aligned} 2a(x_2 - x_1) + 2b(y_2 - y_1) + 2c(z_2 - z_1) + (x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2) &= 0 \\ 2a(x_3 - x_1) + 2b(y_3 - y_1) + 2c(z_3 - z_1) + (x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2) &= 0 \\ 2a(x_3 - x_2) + 2b(y_3 - y_2) + 2c(z_3 - z_2) + (x_2^2 - x_3^2 + y_2^2 - y_3^2 + z_2^2 - z_3^2) &= 0 \end{aligned} \tag{7}$$

$$\begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) & 2(z_2 - z_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 2(z_3 - z_1) \\ 2(x_3 - x_2) & 2(y_3 - y_2) & 2(z_3 - z_2) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} x_1^2 - x_2^2 + y_1^2 - y_2^2 + z_1^2 - z_2^2 \\ x_1^2 - x_3^2 + y_1^2 - y_3^2 + z_1^2 - z_3^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + z_2^2 - z_3^2 \end{bmatrix} \tag{8}$$

$$\begin{aligned} r^2 &= (x_1 - a)^2 + (y_1 - b)^2 + (z_1 - c)^2 \\ r^2 &= (x_2 - a)^2 + (y_2 - b)^2 + (z_2 - c)^2 \\ r^2 &= (x_3 - a)^2 + (y_3 - b)^2 + (z_3 - c)^2 \end{aligned} \tag{9}$$

Search for the sphere in the depth layer image requires its data to be projected from 2D coordinates to 3D points. Furthermore, a depth parameter *depth unit* from the camera settings is required to transform depth units into meters. In the realsense D435 a default value of 0.001 has been found as the *depth unit* value. The depth layer is composed of 16-bit integers. The intrinsic matrix found in phase one of the proposed method is used for projects 2D data to 3D points as follows:

- Convert from integer data in depth image I to depth in meters Z requires multiplying each integer value in the depth image by the depth unit:

$$Z(I_{ij})_{ij} = I_{ij} \times 0.001 \tag{10}$$

- Project from 2D coordinates to 3D world coordinates (Note undistortion has already been applied in phase one):

$$\begin{aligned} X(I_{ij})_{ij} &= (i - c_x) / f_x \times 0.001 \\ Y(I_{ij})_{ij} &= (i - c_y) / f_y \times 0.001 \end{aligned} \tag{11}$$

- 3D cloud point is represented as (X, Y, Z) coordinates.

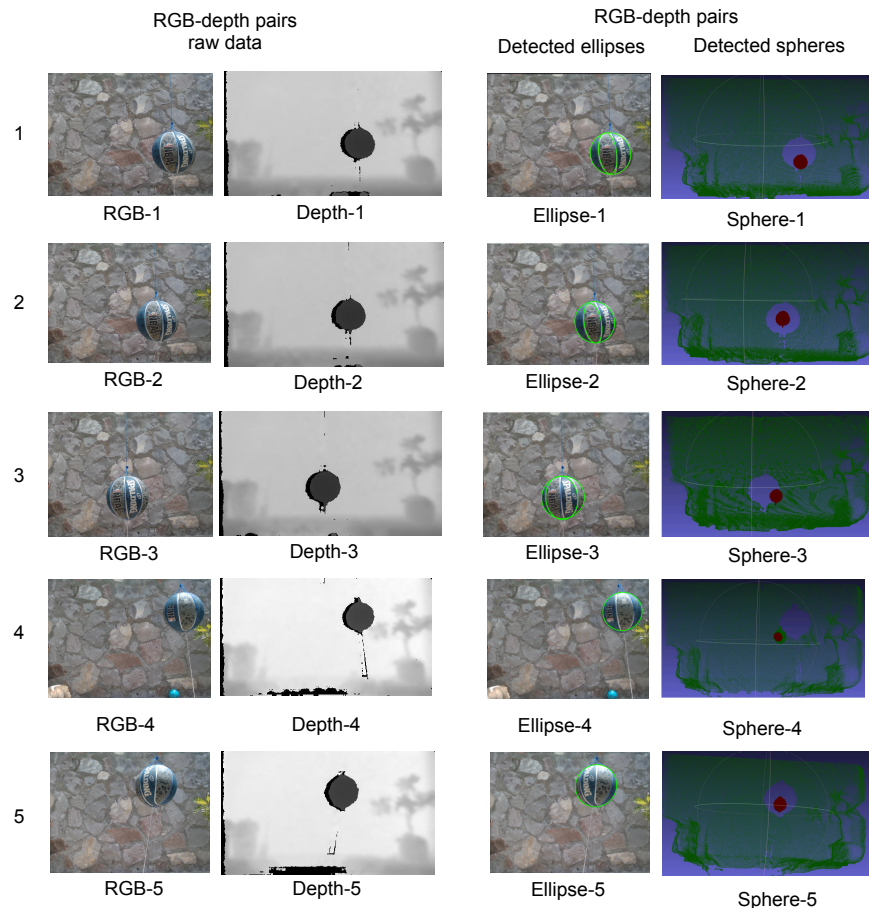


Figure 9. A set of five different images with a basketball visible in the scene is needed, images from 1 to 5 show the same scene with a basketball in 5 different positions, before (left) and after (right) detection of ellipse centers and spheres centers.

Searching for the sphere center and their surface 3D points, we propose a method using RANSAC in Algorithm 1, where the known size of the basketball applies to distinguish it from similar geometries in the scene.

In phase four in Figure 8, having five pairs of ellipse centers and sphere centers, Quest method is incorporated into our methodology to find the rotation and translation matrices. This method has better accuracy in the presence of noise and allows us to use a minimum number of samples in the calibration protocol [41]. Furthermore, Quest method has been used to find rotation and translation from a pair of different views of the same scene with an RGB image and a depth image. Each layer requires its intrinsic matrix, and sphere centers in the depth layer were de-projected to a euclidean space with the intrinsic matrix of the depth layer, as the Quest method implementation needs to work with the same camera. A second alternative is to de-project world coordinate sphere centers, created from the depth layer in a 3D cloud, to a virtual view with the same RGB intrinsic matrix.

2.4. Calibration Protocol

With the proposed method, the calibration process has been simplified to the following steps:

1. At least ten valid chessboard color images are sampled to get intrinsics matrix and undistort RGB images;
2. At least five good pairs of color and depth images are sampled to search for ellipses in the color layer with [46] and spheres in the depth layer with Algorithm 1;
3. Intrinsic matrix, ellipse centers, and sphere centers feed the QuEst method [41] to get rotation and translation matrices.

Algorithm 1 Fit sphere with a known size using RANSAC in a 3D cloudpoint.

Initialize variables to search for a basketball of a standard size of 7 (about 24 cm in diameter and radius of 12 cm). Epsilon is configured as a tolerance for the sphere geometry to 1.2 cm, and stop criteria as an iteration limit of 100,000. All these variables were defined using a heuristic to fit a sphere of known size.

Require: $\epsilon \geq 0.012$

Ensure: $stopcriteria \geq 100,000$

Ensure: $radius \geq 0.12$

Ensure: $diameter \leftarrow 0.024$

1: Select three points from the 3D world coordinate space required by the equations system.

(a).—The first point is selected randomly from the entire 3D cloud space.

(b).—The second and third points are selected reducing the search area within a maximum distance in meters diameter from the first selected point.

2: Solve for the parameters in Equation (8) in the form $Ax = B$,

if an sphere center is found **then**.

3: Determine if the three points and the sphere center fit with a predefined tolerance ϵ w.r.t. the radius.

if yes then

The center of the fitted sphere is added to a list of sphere candidates

Iterate to step 1 a defined number of times $stopcriteria$.

else if no then

Iterate to step 1 a defined number of times $stopcriteria$.

end if

else if no then

Check if $stopcriteria$ has been reach.

if yes then

continue to step 4.

else if no then

continue iterating step 3.

end if

end if

4.—Iterate all points in the 3D coordinate space and check if each point is an inlier with a tolerance ϵ in a $radius$ distance within each point in the list of sphere candidates.

if yes then

This point is added to a list attached to the center where it fits.

end if

5: Check if the list of sphere candidates is not empty,

if yes then

No spheres has been found, if there is a sphere in the scene the method needs to be executed again.

go to step 1

else if no then

Iterate the list of sphere candidates.

Select the center with the more significant amount of points in its attached list of sphere points as the best sphere candidate to fit the basketball geometry of known size.

end if

6: Sphere center and 3D points in the sphere surface have been found.

3. Results

The experiments consider our methodology and [36] method using their available toolbox [42]. In the experiments, our comparison stress the Staranovics method using a minimum of images where their available toolkit could produce an output. Taking one pair of depth and RGB images requires sampling each image consecutively, and the basketball needs to be kept steady with both methods. Furthermore, we use [43] method to undistort RGB images with [42] and our method as a mandatory prerequisite and undistortion of the RGB images benefit results in both methods. It is essential to note that Staranovics ideal number of image pair samples is above 130 to produce much better results than the reported in this work, at the cost of increasing the calibration protocol complexity. When comparing calibration operator tasks, each image pair requires manual intervention in the [42] to ensure correct ellipse and sphere detection. Moreover our methodology uses [46] and their toolkit [46] to find ellipses in the complete image, and our RANSAC method to find the basketball sphere does not require a mandatory step to reduce the search area to fit the sphere geometry in the scene. Furthermore, both methods require valid images where ellipses and spheres are well detected. Not all pair images are good candidates; we sample more pair images than needed to select a minimum of valid images in both methods. Moreover, our proposed method requires fewer steps to choose the correct depth and color images pair. The proposed calibration protocol experiments use a non-ideal scene with natural indirect illumination from the sun, with non-regular textures in the background. Our method shows an emergent resilience behavior as a product of [46], and our RANSAC method to detect ellipses and spheres.

Default RGB camera intrinsic parameter values are shown in Table 2, our work and Staravonics method require intrinsic parameter values. Furthermore, Table 2 shows that default parameters are not close to precision and exactitude than well-established methods such as [36,43]. Nevertheless, Staranowicz's method [36] produces, as a result, another intrinsic RGB matrix, besides RGB input images are pre-calibrated with [43] as in our method.

Rotation and translation factory values are shown in Table 3 and Table 4, Staranowicz's method [36] produces close results to the factory values, and the QUEsT method [41] values are shown as a scale.

Table 2. Color camera (RGB) intrinsics parameters.

| Parameter | F_x | F_y | C_x | C_y | Skew |
|--------------------|-----------|-----------|-----------|-----------|-----------|
| Factory RGB | 421.46 | 421.46 | 461.683 | 236.524 | 0.0 |
| Zhang et al. | 556.28364 | 555.65570 | 325.71797 | 253.41162 | 0.0 |
| Staranowicz et al. | 549.1128 | 550.8689 | 322.2678 | 248.1722 | −2.126733 |

Table 3. Depth camera (RGB-D) rotation parameters.

| Parameter | X | Y | Z | W |
|--------------------|------------|------------|------------|------------|
| Factory | 0.0005242 | 0.0002236 | 0.0044973 | −0.9999897 |
| Fathian et al. | 0.4188 | 0.6221 | 0.5903 | −0.2985 |
| Staranowicz et al. | −0.0089974 | −0.0025626 | −0.0029216 | 0.999952 |

Table 4. Depth camera (RGB-D) translation parameters.

| Parameter | X | Y | Z |
|--------------------|-----------|--------------|-------------|
| Factory | 0.0146951 | −0.000142742 | 0.000202387 |
| Fathian et al. | −0.0811 | 0.2968 | 0.4095 |
| Staranowicz et al. | 0.013768 | −0.0098439 | 0.011041 |

Our proposed method requires to use of a interpolation factor value to approximate the projection of 3D cloud points to a 2D image. We use a cubic spline regression in Figure 10 to interpolate depth error with the scale values produced by the QUEsT method [41].

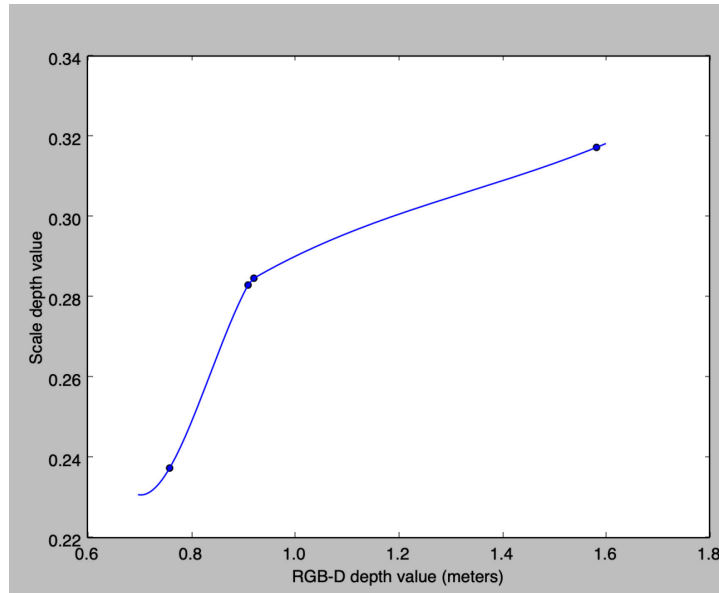


Figure 10. In the figure, the depth values provided by the QUEsT method [41] are fitted with a spline regression. The fitted line shows a non-uniform displacement of the RGB-D depth values [49].

To compare our methodology qualitatively, we perform a reprojection of the center of a sphere, in 3D world coordinates, to the RGB image 2d space; and with the sphere’s radius, draw a red circle as shown in Figure 11b. Our method produces competitive results with [36] (Figure 11a) with fewer pair images.

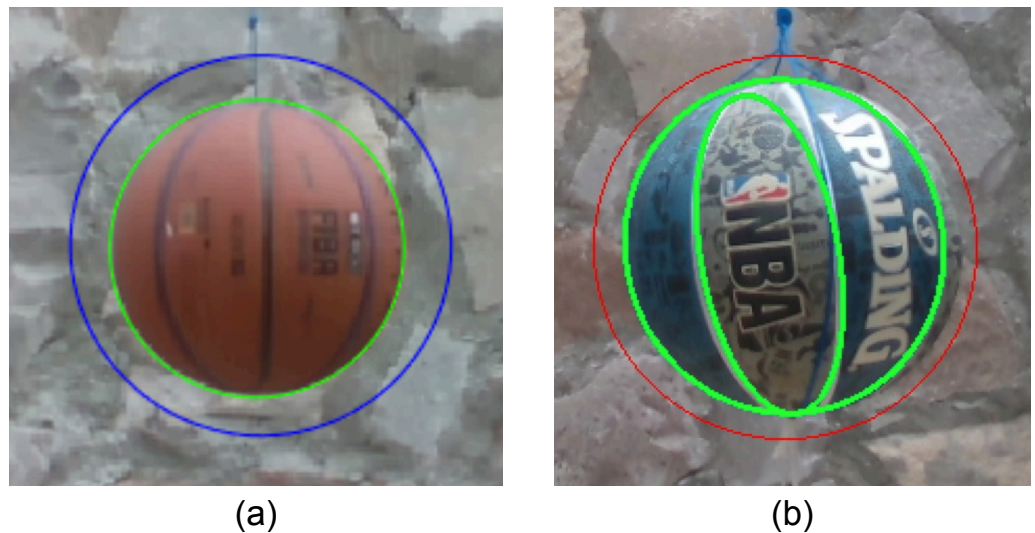


Figure 11. Projection result of the detected sphere using [42] in blue color (a); projection result with our proposed method (b).

Reprojection error and amount of color-depth samples differences are shown in Table 5. Calibration methods using checkerboards in their entire pipeline [33,50] require more samples than our work, as the checkerboard squares are not visible in the depth layer. These methods rely on detecting corners.

Table 5. Re-Projection error and amount of samples.

| Method | Reprojection Error (Pixels) | Color-Depth Pair Samples |
|-------------------------|-----------------------------|--------------------------|
| Herrera et al. [33] | 2.388 | 20–60 |
| Staranowicz et al. [35] | 4.8248 | 25–120 |
| Basso et al. [50] | 1.901 | above 100 |
| Zhou et al. [40] | 0.257039 | 40 |
| Our method | 0.0768 | 5 |

4. Discussion

In the final phase of our method (Figure 8), QUEsT method [41] and the available toolbox [51] consider two views from the same scene with only one intrinsic camera parameter matrix.

Our proposed methodology uses depth information in a 3D cloud point as 3D world coordinates. Then, sphere centers are projected to a 2D view using depth layer intrinsic parameter matrix. Alternatively, the proposed method can project 3D world coordinates to a virtual view with RGB camera intrinsics. The QUEsT method [41] produces depth, rotation, and translation (up to a scale factor), * Italics have been removed and accurate 3D world coordinates are required to translate scale depth values into meters. Our methodology use depth information to create 3D world coordinates. Moreover, depth data shows a depth error. As a general rule in RGB-D cameras, depth error increases when the camera's distance from the objects in the scene is not uniform in the depth image volume [49]. Future work will adjust this calibration error separately. In the current work, we use a spline regression to translate and interpolate scale depth values to meters.

The proposed methodology reduces the complexity of the calibration protocol for RGB-D cameras by minimizing the number of samples that the calibration protocol operator needs to take. It allows non-ideal illumination conditions in the scene and complex textures in the scene. It reduces the effort operator needs to invest in performing the calibration protocol. Furthermore, the proposed method shows competitive results in a non-ideal scene, and experiments demonstrate that a straightforward calibration protocol is possible, requiring a minimum number of samples using a sphere of know size as a geometrical object in the scene and incorporating state-of-the-art techniques in our methodology.

Author Contributions: Methodology, L.-R.R.-R. and J.C.P.-O.; software, I.S.-R.; validation, J.M.R.-A.; formal analysis, M.A.A.-F.; investigation, L.-R.R.-R. and I.S.-R.; writing—original draft preparation, L.-R.R.-R. and J.C.P.-O.; writing—review and editing, E.G.-H. and M.A.A.-F.; supervision, J.C.P.-O.; project administration, J.M.R.-A. and L.-R.R.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We acknowledge Fondo para el fomento de la cultura emprendedora de la faculta de ingeniería 2020, Universidad Autónoma de Querétaro for their support to help this project to reach positive results.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Slavcheva, M.; Baust, M.; Cremers, D.; Ilic, S. Killingfusion: Non-rigid 3d reconstruction without correspondences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1386–1395.
- Huang, X.; Zhang, Y.; Xiong, Z. High-speed structured light based 3D scanning using an event camera. *Opt. Express* **2021**, *29*, 35864–35876. [[CrossRef](#)] [[PubMed](#)]

3. Han, J.; Shao, L.; Xu, D.; Shotton, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Trans. Cybern.* **2013**, *43*, 1318–1334. [[PubMed](#)]
4. Giancola, S.; Valenti, M.; Sala, R. *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*; Springer: Berlin/Heidelberg, Germany, 2018.
5. Keselman, L.; Iselin Woodfill, J.; Grunnet-Jepsen, A.; Bhowmik, A. Intel realsense stereoscopic depth cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 1–10.
6. Silva Neto, J.; Lima Silva, P.; Figueredo, F.; Teixeira, J.; Teichrieb, V. Comparison of RGB-D sensors for 3D reconstruction. In Proceedings of the 2020 22nd Symposium On Virtual And Augmented Reality (SVR), Porto de Galinhas, Brazil, 7–10 November 2020; pp. 252–261.
7. Zollhöfer, M. Commodity RGB-D sensors: Data acquisition. In *RGB-D Image Analysis and Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 3–13.
8. Neupane, C.; Koirala, A.; Wang, Z.; Walsh, K.B. Evaluation of depth cameras for use in fruit localization and sizing: Finding a successor to kinect v2. *Agronomy* **2021**, *11*, 1780. [[CrossRef](#)]
9. LeCompte, M.C.; Chung, S.A.; McKee, M.M.; Marshall, T.G.; Frizzell, B.; Parker, M.; Blackstock, A.W.; Farris, M.K. Simple and Rapid Creation of Customized 3-dimensional Printed Bolus Using iPhone X True Depth Camera. *Pract. Radiat. Oncol.* **2019**, *9*, e417–e421. [[CrossRef](#)] [[PubMed](#)]
10. Tagarakis, A.C.; Kalaitzidis, D.; Filippou, E.; Benos, L.; Bochtis, D. 3D Scenery Construction of Agricultural Environments for Robotics Awareness. In *Information and Communication Technologies for Agriculture—Theme III: Decision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 125–142.
11. Sui, W.; Wang, L.; Fan, B.; Xiao, H.; Wu, H.; Pan, C. Layer-wise floorplan extraction for automatic urban building reconstruction. *IEEE Trans. Vis. Comput. Graph.* **2015**, *22*, 1261–1277. [[CrossRef](#)] [[PubMed](#)]
12. Klingensmith, M.; Dryanovski, I.; Srinivasa, S.S.; Xiao, J. Chisel: Real Time Large Scale 3D Reconstruction Onboard a Mobile Device using Spatially Hashed Signed Distance Fields. In *Robotics: Science and Systems*; Citeseer: Princeton, NJ, USA, 2015; Volume 4.
13. Fu, Y.; Yan, Q.; Yang, L.; Liao, J.; Xiao, C. Texture mapping for 3d reconstruction with rgb-d sensor. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4645–4653.
14. Zollhöfer, M.; Stotko, P.; Görlitz, A.; Theobalt, C.; Nießner, M.; Klein, R.; Kolb, A. State of the art on 3D reconstruction with RGB-D cameras. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2018; Volume 37, pp. 625–652.
15. Yuan, Z.; Li, Y.; Tang, S.; Li, M.; Guo, R.; Wang, W. A survey on indoor 3D modeling and applications via RGB-D devices. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 815–826. [[CrossRef](#)]
16. Li, J.; Gao, W.; Wu, Y.; Liu, Y.; Shen, Y. High-quality indoor scene 3D reconstruction with RGB-D cameras: A brief review. *Comput. Vis. Media* **2022**, *8*, 369–393. [[CrossRef](#)]
17. Chidsin, W.; Gu, Y.; Goncharenko, I. AR-based navigation using RGB-D camera and hybrid map. *Sustainability* **2021**, *13*, 5585. [[CrossRef](#)]
18. Song, Y.; Xu, F.; Yao, Q.; Liu, J.; Yang, S. Navigation algorithm based on semantic segmentation in wheat fields using an RGB-D camera. *Inf. Processing Agric.* **2022**. [[CrossRef](#)]
19. Antonopoulos, A.; Lagoudakis, M.G.; Partsinevelos, P. A ROS Multi-Tier UAV Localization Module Based on GNSS, Inertial and Visual-Depth Data. *Drones* **2022**, *6*, 135. [[CrossRef](#)]
20. Wang, F.; Zhang, C.; Zhang, W.; Fang, C.; Xia, Y.; Liu, Y.; Dong, H. Object-Based Reliable Visual Navigation for Mobile Robot. *Sensors* **2022**, *22*, 2387. [[CrossRef](#)] [[PubMed](#)]
21. Morell-Gimenez, V.; Saval-Calvo, M.; Azorin-Lopez, J.; Garcia-Rodriguez, J.; Cazorla, M.; Orts-Escolano, S.; Fuster-Guillo, A. A comparative study of registration methods for RGB-D video of static scenes. *Sensors* **2014**, *14*, 8547–8576. [[CrossRef](#)] [[PubMed](#)]
22. Pan, Y.; Chen, C.; Li, D.; Zhao, Z.; Hong, J. Augmented reality-based robot teleoperation system using RGB-D imaging and attitude teaching device. *Robot. Comput. Integr. Manuf.* **2021**, *71*, 102167. [[CrossRef](#)]
23. Tanzer, M.; Laverdière, C.; Barimani, B.; Hart, A. Augmented Reality in Arthroplasty: An Overview of Clinical Applications, Benefits, and Limitations. *J. Am. Acad. Orthop. Surg.* **2022**, *30*, e760–e768. [[CrossRef](#)]
24. Yu, K.; Eck, U.; Pankratz, F.; Lazarovici, M.; Wilhelm, D.; Navab, N. Duplicated Reality for Co-located Augmented Reality Collaboration. *IEEE Trans. Vis. Comput. Graph.* **2022**, *28*, 2190–2200. [[CrossRef](#)]
25. Oliveira, M.; Santos, V.; Sappa, A.D.; Dias, P.; Moreira, A.P. Incremental texture mapping for autonomous driving. *Robot. Auton. Syst.* **2016**, *84*, 113–128. [[CrossRef](#)]
26. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [[CrossRef](#)]
27. Liu, Z.; Zhao, C.; Wu, X.; Chen, W. An effective 3D shape descriptor for object recognition with RGB-D sensors. *Sensors* **2017**, *17*, 451. [[CrossRef](#)]
28. Na, M.H.; Cho, W.H.; Kim, S.K.; Na, I.S. Automatic Weight Prediction System for Korean Cattle Using Bayesian Ridge Algorithm on RGB-D Image. *Electronics* **2022**, *11*, 1663. [[CrossRef](#)]
29. Tan, F.; Xia, Z.; Ma, Y.; Feng, X. 3D Sensor Based Pedestrian Detection by Integrating Improved HHA Encoding and Two-Branch Feature Fusion. *Remote Sens.* **2022**, *14*, 645. [[CrossRef](#)]

30. Zheng, H.; Wang, W.; Wen, F.; Liu, P. A Complementary Fusion Strategy for RGB-D Face Recognition. In Proceedings of the International Conference on Multimedia Modeling, Phu Quoc, Vietnam, 6–10 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 339–351.
31. Zhang, C.; Zhang, Z. Calibration between depth and color sensors for commodity depth cameras. In *Computer Vision and Machine Learning with RGB-D Sensors*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 47–64.
32. Darwish, W.; Tang, S.; Li, W.; Chen, W. A new calibration method for commercial RGB-D sensors. *Sensors* **2017**, *17*, 1204. [[CrossRef](#)] [[PubMed](#)]
33. Herrera, D.; Kannala, J.; Heikkilä, J. Joint depth and color camera calibration with distortion correction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2058–2064. [[CrossRef](#)] [[PubMed](#)]
34. Basso, F.; Pretto, A.; Menegatti, E. Unsupervised intrinsic and extrinsic calibration of a camera-depth sensor couple. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 6244–6249.
35. Staranowicz, A.; Brown, G.R.; Morbidi, F.; Mariottini, G.L. *Easy-to-Use and Accurate Calibration of RGB-D Cameras from Spheres*; Springer: Berlin/Heidelberg, Germany, 2014.
36. Staranowicz, A.N.; Brown, G.R.; Morbidi, F.; Mariottini, G.L. Practical and accurate calibration of RGB-D cameras using spheres. *Comput. Vis. Image Underst.* **2015**, *137*, 102–114. [[CrossRef](#)]
37. Liu, H.; Li, H.; Liu, X.; Luo, J.; Xie, S.; Sun, Y. A novel method for extrinsic calibration of multiple RGB-D cameras using descriptor-based patterns. *Sensors* **2019**, *19*, 349. [[CrossRef](#)]
38. Chen, C.; Yang, B.; Song, S.; Tian, M.; Li, J.; Dai, W.; Fang, L. Calibrate multiple consumer RGB-D cameras for low-cost and efficient 3D indoor mapping. *Remote Sens.* **2018**, *10*, 328. [[CrossRef](#)]
39. Zhong, J.; Li, M.; Liao, X.; Qin, J. A real-time infrared stereo matching algorithm for RGB-D cameras' indoor 3D perception. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 472. [[CrossRef](#)]
40. Zhou, Y.; Chen, D.; Wu, J.; Huang, M.; Weng, Y. Calibration of RGB-D Camera Using Depth Correction Model. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2203, p. 012032.
41. Fathian, K.; Ramirez-Paredes, J.P.; Doucette, E.A.; Curtis, J.W.; Gans, N.R. Quest: A quaternion-based approach for camera motion estimation from minimal feature points. *IEEE Robot. Autom. Lett.* **2018**, *3*, 857–864. [[CrossRef](#)]
42. Staranowicz, A.N.A. astaranowicz/DCCT: Depth-Camera Calibration Toolbox (RGB-D Calibration ToolBox). GitHub. Available online: <https://github.com/astaranowicz/DCCT> (accessed on 15 June 2022).
43. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [[CrossRef](#)]
44. Enazoe. Enazoe/Camera Calibration cpp: C Detail Implementation of Camera Calibration. Available online: https://github.com/enazoe/camera_calibration_cpp (accessed on 4 January 2022).
45. Burger, W. Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation (Technical report HGB16-05). Available online: https://www.researchgate.net/publication/303233579_Zhang's_Camera_Calibration_Algorithm_In-Depth_Tutorial_and_Implementation (accessed on 4 January 2022).
46. Lu, C.; Xia, S.; Shao, M.; Fu, Y. Arc-support line segments revisited: An efficient high-quality ellipse detection. *IEEE Trans. Image Process.* **2019**, *29*, 768–781. [[CrossRef](#)]
47. Drap, P.; Lefèvre, J. An exact formula for calculating inverse radial lens distortions. *Sensors* **2016**, *16*, 807. [[CrossRef](#)]
48. Tsai, R. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. Robot. Autom.* **1987**, *3*, 323–344. [[CrossRef](#)]
49. Rosin, P. *RGB-D Image Analysis and Processing*; Springer: Cham, Switzerland, 2019.
50. Basso, F.; Menegatti, E.; Pretto, A. Robust intrinsic and extrinsic calibration of RGB-D cameras. *IEEE Trans. Robot.* **2018**, *34*, 1315–1332. [[CrossRef](#)]
51. Kaveh Fathian—QuEst 5-Point. Available online: <https://sites.google.com/view/kavehfathian/code/quest-5-point> (accessed on 4 January 2022).