# Simple and precise multi-view camera calibration for 3D reconstruction

Alberto J. Perez[a,b,*], Juan-Carlos Perez-Cortes[a,b], Jose-Luis Guardiola[a]

[a] *Instituto Tecnológico de Informática (ITI), Universitat Politècnica de València, 46022 Valencia, SPAIN*

[b] *Departamento de Informática de Sistemas y Computadores (DISCA), Universitat Politècnica de València, 46022 Valencia, SPAIN*

## Abstract

A precise calibration in multi-view camera environments allows to perform accurate 3D object reconstruction, precise tracking of objects and accurate pose estimation. Those techniques are of high value in the industry today in fields as quality control or automation. In the present work, an improvement of a simple existing multi-view camera calibration method is presented. The improved method employs a specially developed reference token to overcome some issues in the original algorithm. We prove that the new method overcomes those problems thus attaining a higher accuracy while keeping the process simple and the implementation costs low. This last aspects makes the method interesting for the industry but specially suitable for SMEs typical in traditional sectors.

*Keywords:* 3D reconstruction, camera calibration, extrinsic camera parameters, multi-camera calibration, calibration token, quality control, industrial automation

[⋆] This work has been supported by Generalitat Valenciana through I+D IVACE (Valencian Institute of Business Competitiveness) and FEDER (European Regional Develoopement Fund) under project IMDEEA/2019/86"

[*] Corresponding author

*Email addresses:* aperez@disca.upv.es (Alberto J. Perez), jcperez@iti.upv.es (Juan-Carlos Perez-Cortes), joguagar@iti.upv.es (Jose-Luis Guardiola)

## 1. Introduction

The 3D analysis of scenes is a key aspect in multiple field but it plays an important role in the digital transformation in industry: products digitization for design, online catalogues for costumers, metrology, quality control, computer aid manufacturing (CAM), ... Traditional sectors comprise Small to Medium Enterprises (SMEs) with limited resources to invest in those technologies, then it is very important to keep those processes simple (less specialized staff is required) and with a reasonable cost.

Several methods or devices are available to perform those tasks and could be classified into contact and non-contact devices. Contact methods such scanning probes are slow and need direct physical contact with the object. On the other hand non-contact methods [6] can be active, based on ultrasound, time-of-flight, structured light, ..., or passive, based on the analysis of images of the object taken from different views (stereoscopy, visual hull, ...).

Passive methods based on the convex hull have several advantages but need a very precise calibration of the cameras to achieve accurate results. In [4], a device is presented based on this technique that allows a 3D reconstruction of an object without occlusions. The calibration of the full camera set of this kind of devices is a critical aspect because all the camera views do not overlap, hence is not possible to use planar patterns as in traditional methods [1, 2, 3]. Calibration cameras by pairs can be considered, but those methods show in practice precision and stability problems since errors may accumulate along the computing chain. The use of complex 3D patterns is possible but those objects are complex to build and to process. In [4, 5] a simple method is proposed using a sphere without any special pattern on it as calibration token. This method allows to calibrate simultaneously a set of cameras, but it presents an scale calibration issue that will be addressed in this paper. Several approaches are studied, one of them involving a new reference token that also offers other interesting benefits to the process.

This work is structured as follows. Section 2 studies the calibration method proposed in [4][5] and its scale issue. Section 3 presents a extension to the algorithm that overcomes that problem. Section 4 shows experiments performed with virtual and real data. Finally, section 5 exposes the conclusions.

## 2. Multi-view calibration based on a spherical token

The procedure described in [4] uses a sphere as reference token to calibrate the extrinsic parameters of a set of cameras. The cameras are distributed on an polyhedral structure pointing to its center. The sphere is presented to the cameras at different locations by simply letting it fall through the device several times. Each time, a synchronized set of images is captured.

As an advantage, a sphere offers the same silhouette from different angles and it can be easily segmented in an image. Furthermore, the center of mass of the obtained blob can be used as an accurate approximation of the sphere center projection if the sphere radius is small compared to the working distance to the camera [8]. Thus, a set of 2D projections (one per each camera) of the same 3D point (the sphere center) is obtained for each falling sphere.

Given a set of 2D points, representing the same 3D world point, a set of camera projection rays (epipolar lines) can be computed. If the cameras are correctly calibrated and the world coordinate system is consistent, all those rays should converge to the same 3D world point. On any other case the rays will not (see Figure 1).

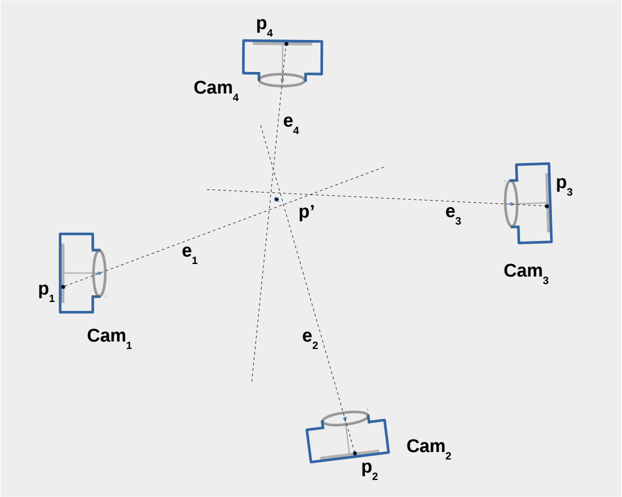

Figure 1: Camera projection rays ($e_i$) of a set of 2D points ($p_i$) representing the same 3D world point in a poorly calibrated system. A point $p'$, that minimizes the distances to all $e_i$, can be used as an estimation of the real point.

**Algorithm 1** Multi-camera extrinsic parameter calibration

```
Require: $M$ capture sets
repeat
    for each captured sphere $S_m$ do
        for each $i$-th camera do
            Let $p_{mi}$ be the center of $S_m$ projected to the camera image
            Compute epipolar line $e_{mi}$
        end for
        $p'_m \leftarrow \arg\min_p \{dist(p, \{e_{mi}\})\}$
        Set $p'_m$ as the corresponding 3D point of $p_{mi}$ 2D points
    end for
    $g_e \leftarrow mean\{dist(p'_m, \{e_{mi}\})\}$
    for each $i$-th camera do
        Recalibrate extrinsic parameters using pairs $\{(p_{mi}, p'_m) : m = 1...M\}$.
    end for
until Stop Criterion
```

Considering a system with $c$ cameras and starting with a rough estimation of the extrinsic parameters, the calibration process in [4] assign to each set of 2D projections obtained from the sphere center projections, $\{p_i : i = 1...c\}$, the 3D world point $p'$, that minimizes the distance to all the camera projection rays (see Figure 1). A closed-form solution to compute this point can be found in [4].

So, Algorithm 1 is proposed to calibrate the extrinsic parameters of the cameras using $M$ capture sets of an sphere.

In [4] the extrinsic parameters are recalibrated using a gradient descent algorithm. The method stops when a minimum value of the global error mean, $g_e$, computed in each iteration is found or a maximum number of iterations is attained. The algorithm requires at least six sets of sphere captures not laying on the same plane given that the number of parameters needed is six for each camera (3D position and 3D rotation).

Figure 2 presents the evolution of Algorithm 1 for the camera configuration described on [4] (see Figure 10), In this execution, 10 capture sets were used. Figure 2 shows the global error mean, the mean distance from the cameras to the centroid computed from their positions, and the mean reciprocal error ($r_e$). This value is obtained for each $i$-th camera as the mean quadratic error between the projections of the $p'_m$ points on the camera sensor using the current calibration parameters and its corresponding $p_{mi}$ points. This measures how well the extrinsic parameters match for a camera.

From Figure 2, we can see how the global error mean decays fast in the first iterations and then evolves slowly. Nevertheless, seeing the evolution of the centroid distance, we can see how the cameras collapse to its centroid, while the retroprojection error stabilizes. Stopping the algorithm at different iterations gives consistent world coordinate systems but with cameras positioned at different distances from the centroid. Thus, a scale calibration issue shows up. The later the algorithm stops, the smaller the objects will be estimated (see Figure 3).

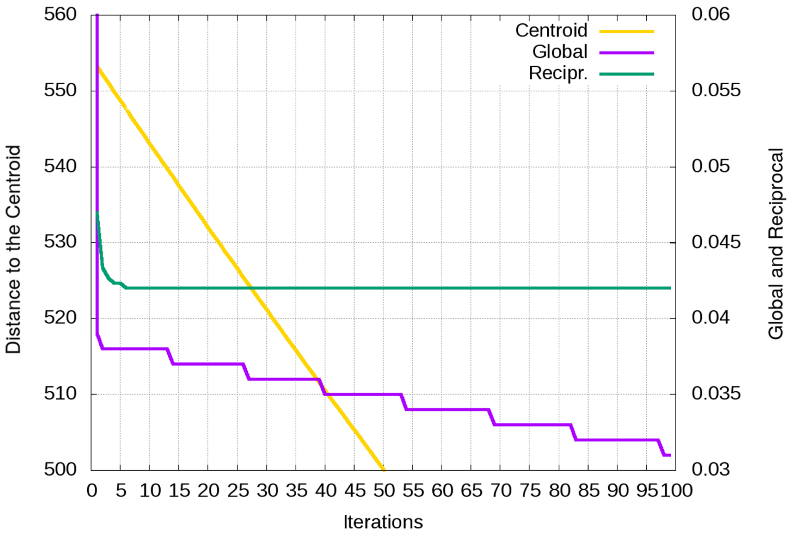

Figure 2: Evolution of the global error mean, to the average centroid distance and the reciprocal error mean during an execution of Algorithm 1.

We can conclude that the algorithm converges for the camera rotation parameters, since the cameras point to the right direction, but their positions collapse to the centroid. This is due to the fact that we are using the sphere centers as calibration points, they have no dimension, then the scale can not be induced correctly from them. Errors can be minimized reducing the distance between the calibration points.

In [4] the authors calibrate the scale reconstructing a sphere of known dimension. They use the calibration obtained by the algorithm and apply a shape-from-silhouette method to perform the reconstruction. The diameter of the reconstructed sphere is estimated and then the cameras are moved away from the centroid to match de correct scale. This method has two drawbacks: 1) a postprocess is required, and 2) the diameter estimation is not accurate enough because this reconstruction technique is affected by small segmentation errors and introduces bumps in the model.

Other possibility is to take into account the apparent area of the sphere in the images while recalibrating the extrinsic parameters using the descent gradient algorithm, but here the segmentation method applied to compute this area may not be accurate enough.

In this work we propose the use a bit more sophisticated calibration token, from which an scale estimation independent of the segmentation process can be performed. We discuss our proposal in the next section.

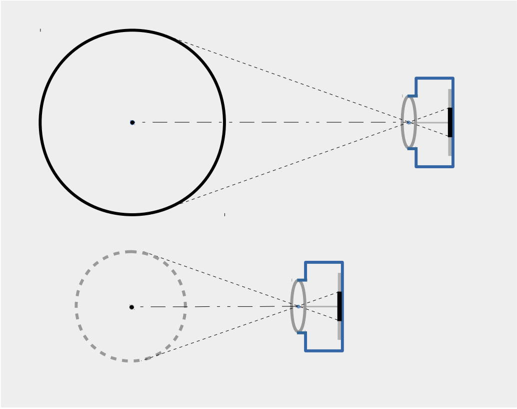

Figure 3: The real camera (top) obtains an image of the object (sphere). If the camera position is improperly estimated (bottom), the object dimension will be incorrectly computed.

## 3. Multi-view calibration based on a multispherical token

To overcome the scale issue described in the previous section we propose a new token from which the scale can be inferred. The token (see Figure 4) is composed by two spheres connected by a rod of fixed length. The spheres are of different sizes to be clearly identified on the images.

Using this token we have two spheres in each image, then we obtain two calibration points per image, and the distance in the world coordinate system

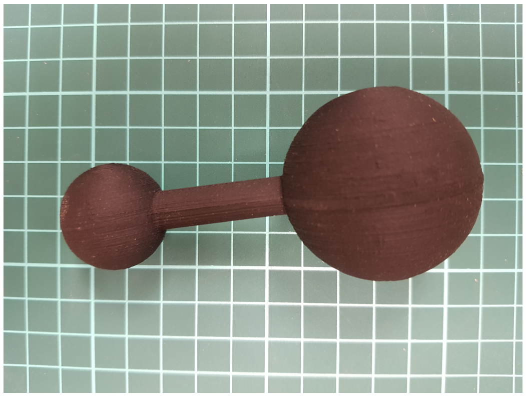

Figure 4: The new token consists on two spheres connected by a rod of fixed length.

between both points is known. Because the blob center used to estimate the sphere center projection is independent from possible erosions and dilations due to the segmentation algorithm, the distance of both points in the image is independent too. This is an advantage not shared by the estimation of the sphere areas.

With this new information, we propose an extension of Algorithm 1 to take into account this new info to avoid the scale calibration issue.

We introduce Algorithm 2 to correct the position of a camera in order to obtain the correct distance between a pair of points that are known to be $d_r$ apart. Figure 5 illustrates the procedure.

With the procedure defined by Algorithm 2, we propose Algorithm 3 as an extension of the original algorithm. On the one hand the algorithm is adapted to take into account that there are two reference points per image, one per sphere of the token. On the other hand a scale adjustment based on the correction of one of the extrinsic parameters, the camera position, is added.

**Algorithm 2** Camera position correction.

```
Being $P$ and $Q$ two 3D points that are seen by a camera with optical center located in position $C$
Compute $\overline{PQ}$ as the segment joining $P$ and $Q$
Compute $M_P$ as the midpoint of $\overline{PQ}$
Compute epipolar lines $e_P$ and $e_Q$ as the lines passing respectively through the points $P$ and $Q$ and the optical center $C$
Compute $\overline{P'Q'}$ as a segment satisfying: 1) being on the same line defined by $\overline{PQ}$, 2) having a size of $d_r$ units, and 3) sharing the midpoint with segment $\overline{PQ}$ ($M_P$)
Compute $e'_P$ as a line parallel to $e_P$ passing through point $P'$, and $e'_Q$ as a line parallel to $e_Q$ passing through point $Q'$
Compute point $C'$ as the corrected optical center as the intersection of lines $e'_P$ and $e'_Q$
```

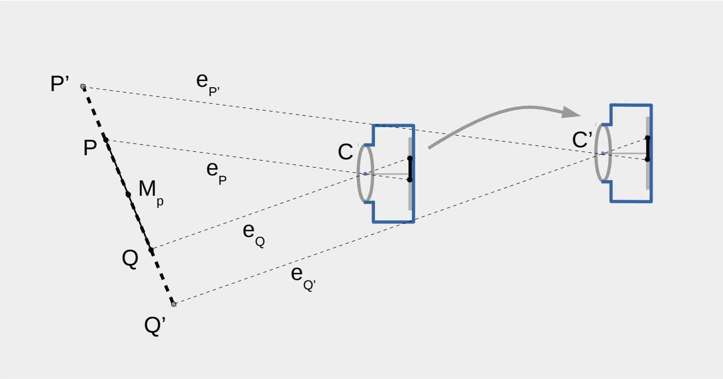

Figure 5: Scale correction applied by Algorithm 2. Camera center is moved to correct the size of segment $\overline{PQ}$ in world coordinates.

The algorithm starts adjusting the extrinsic parameters as in the original one. After convergence, a step of scale correction is added and the camera positions are modified to match the correct scale. The algorithm repeats this procedure until convergence of the scale corrections. In the original algorithm, the extrinsic parameter convergence was determined by the global error mean, $g_e$, but analyzing the results, we consider that the convergence of the reciprocal error mean, $r_e$, is more appropriate since it converges faster (see Figure 2).

**Algorithm 3** Multi-camera extrinsic parameter calibration with scale correction

```
Require: $M$ capture sets
repeat
    # Extrinsic parameters #
    repeat
        for each captured token with spheres $S_{m_1}$ and $S_{m_2}$ do
            for each $i$-th camera do
                Let $p_{m_1 i}$ and $p_{m_2 i}$ be the centers of the spheres projected to the
                camera image
                Compute epipolar lines $e_{m_1 i}$ and $e_{m_2 i}$
            end for
            $p'_{m_1} \leftarrow \arg\min_{p}\{dist(p, \{e_{m_1 i}\})\}$
            $p'_{m_2} \leftarrow \arg\min_{p}\{dist(p, \{e_{m_2 i}\})\}$
            Set $p'_{m_1}$ as the corresponding 3D point of $\{p_{m_1 i}\}$ 2D points
            Set $p'_{m_2}$ as the corresponding 3D point of $\{p_{m_2 i}\}$ 2D points
        end for
        $g_e \leftarrow mean\{dist(p'_{m_s}, \{e_{m_s i}\})\}$
        for each $i$-th camera do
            Recalibrate extrinsic parameters using pairs $\{(p_{m_s i}, p'_{m_s})\}$.
            Compute reciprocal error, $r_{e_i}$
        end for
        $r_e \leftarrow mean\{r_{e_i}\}$
    until $r_e$ convergence
    # Scale correction #
    for each $i$-th camera with optical center $C_i$ do
        Set $e_i$ to 0
        for each $(p, q) \in \{(p'_{m_1}, p'_{m_2})\}$ do
            Compute $C'_i$ by means of Algorithm 2
            $e_i \leftarrow e_i + (C'_i - C_i)$
        end for
        $C_i \leftarrow C_i + \frac{e_i}{M}$
    end for
    $e \leftarrow mean\{e_i\}$
until $e$ convergence
```

A disadvantage of using the new token is that the estimation of the projection of sphere centers is more complex. With a sphere, it is straightforward to obtain an estimation of its center projection just by computing the center of mass of the blob obtained in the image. In the case of the double sphere token, its segmented blob contains both spheres, thus the estimation of its center projection have to be done differently. Moreover, possible occlusions have to be taken into account. In section 3.1, we present a method to compute accurately the sphere center projections and to deal properly with occlusions.

Finally, besides giving a scale reference, the new token provides additional advantages. On the one hand it provides more reference points per capture set, and on the other hand it offers more variate sights in the captures because it tends to rotate while falling. This reduces the probability that the reference points lay on the same plane, hence allowing a more accurate calibration.

### *3.1. Sphere center projection*

As stated before, the computation of sphere center projections with the new token is not as straightforward as with spherical tokens. Besides spheres can be occluded in some of the captures (see Figure 6), thus the sphere center projection may not be computable and the corresponding images have to be discarded. In this section we present an algorithm to perform those computations.

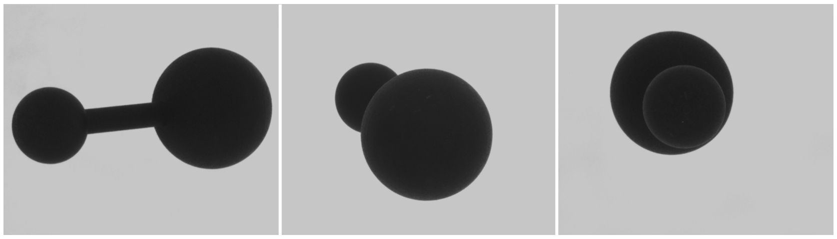

Figure 6: Possible views of a double sphere token: full view of both spheres (left), partial occlusion (middle), total occlusion (right).

First, each image is binarized using a thresholding technique [10] and a set of points containing the contour of the token is created. Next, the major axis is computed applying principal component analysis (PCA, [11]) to the point set. The more significative PCA projection is used as the main axis (Figure 7). After that, the mean profile is computed using profile points from both sides of the major axis (Figure 7). The maxima ($m_0$, $m_1$) in this profile are roughly the sphere centers, that are calculated locating the zero crossings on the smoothed derivative profile. The minimum ($m$) between the maxima is computed too.

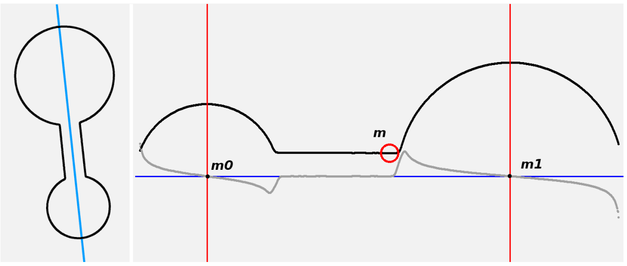

Figure 7: Left: token blob contour and major axis. Right: profile graph (black line) and derivative profile graph (gray line). Points $m_o$ and $m_1$ are the profile maxima, point $m$ is the minimal point between them.

This first center estimation is refined by a procedure that splits the contour set into two subset of points that initially contain points from each side of the minimum ($m$). Then an ellipse fitting algorithm [9] is applied to each subset and points outside the fitted ellipses are removed (Figure 8a). This process is repeated until no new points are removed and the ellipse centers (in this case, circles) are provided as center estimations. This procedure offers an more robust estimation that just the maxima computed in the first step.

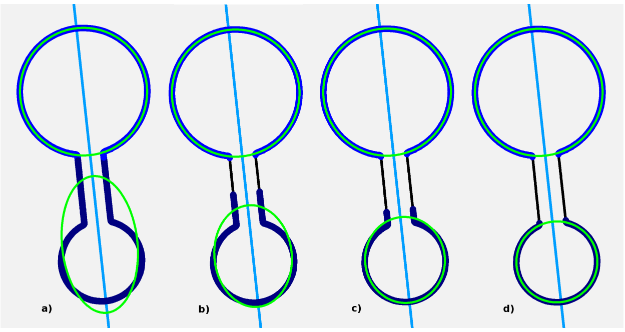

Figure 8: From a) to d): iterative sphere location adjustment.

In the case of captures with occlusions (Figure 9) the initial subset partition will provide directly the center estimation, no iteration is performed.

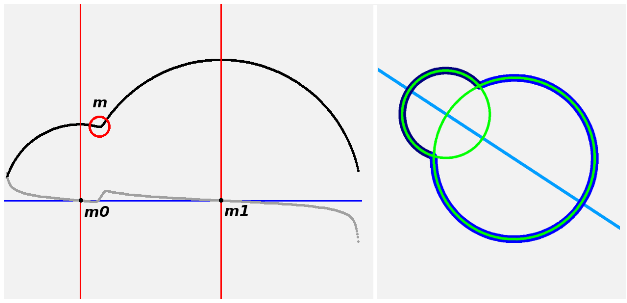

Figure 9: Left: profile graph of a token with occlusion. Right: direct sphere center location.

## 4. Experiments and results

A rendering software tool has been developed to simulate a device with 16 cameras distributed as in Figure 10. This software matchs the configuration of a device in our laboratory [4]. Each camera has a focal length of 25 mm and a sensor size of 2448x2048 pixels. The cameras are distributed inside a 550 mm radius sphere. A virtual calibration token was also designed matching our existing calibrated token (two spheres 65.25 mm apart with radius 43.5 and 26.1 mm)

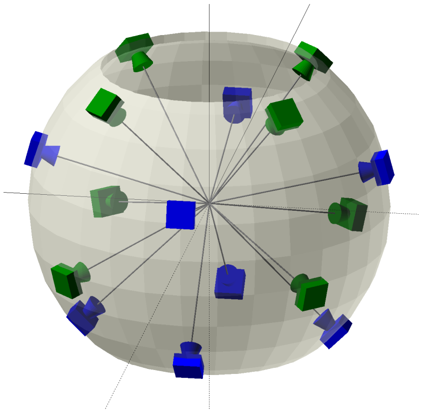

Figure 10: Camera configuration. A set of 16 cameras distributed over a spherical surface, all them pointing to its center.

Several experiments have been performed with the real and the virtual devices to test the developed algorithms. On the one hand, five synthetic calibration sets composed each by 20 capture sets have been created. In each capture set the token has been positioned randomly near the device center and 16 synthetics views (one for each camera) have been obtained. A total of $5 \times 20 \times 16 = 1600$ images have been generated. On the other hand, five real calibration sets composed by 20 capture sets have been obtained capturing the calibration token with the physical device. This device synchronizes the capture of the 16 cameras to obtain 16 views of an object as it is falling (more details in [4]).

To test the sphere center projection algorithm in the virtual sets, since the 3D position of the token on each simulation is known, a ground truth is generated computing the exact position of the sphere center projections on the images according to the virtual camera model. With this information, the accuracy of the algorithm in ideal conditions is tested giving an upper bound of its performance. From the original 1600 captures, a total of 1482 valid images have been obtained, of which 118 present light occlusions. The rest have been discarded because too large occlusions were found. The occlusion threshold can be set arbitrary. In our case, the light occlusion limit is defined as an occlusion where the center of the little sphere lays at the edge of the big sphere.

As can be seen in Figure 11 and Table 1, the error estimation is below one image pixel. Predictably, in the case of the small sphere the estimation is more accurate. This is because the sphere center projection is estimated as the center of mass of the blob generated by the sphere on the image, and those values are known to deviate more as the sphere radius increases [8].

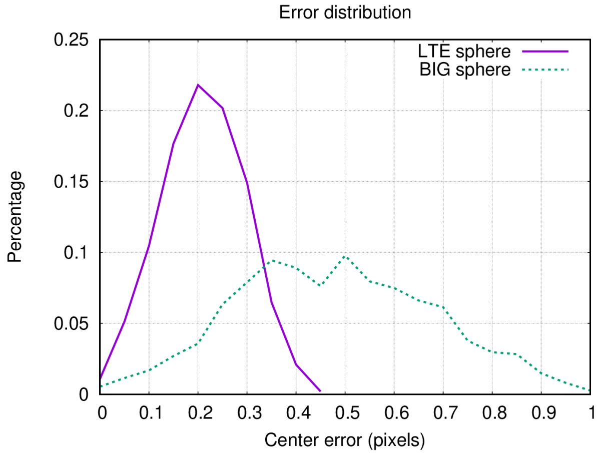

Figure 11: Distribution of the estimation errors of center projection for both token spheres.

Table 1: Estimation error statistics of center projections for both spheres.

| **Sphere** | **Mean** | **Std. deviation** | **Maximum** |
|---|---|---|---|
| *big* | 0.505 | 0.202 | 1.045 |
| *little* | 0.235 | 0.085 | 0.479 |

To test the convergence of the new calibration algorithm, the virtual and the real calibration sets have been employed. The algorithm uses the estimated sphere center projections to proceed. Nevertheless, in the case of the virtual sets, the exact sphere center projections, computed for the ground truth of the previous experiment, can be used to establish the influence of the accuracy of this estimation in the convergence. Considering this, three experiment setups have been designed,

- Convergence with virtual sets using exact sphere center projections
- Convergence with virtual sets using estimated sphere center projections
- Convergence with real sets using estimated sphere center projections

As stated in section 2, a rough estimation of the external parameters is needed to run the algorithm. For the virtual sets, these are extracted from a modified version of the cameras model employed to generate the captures. In this version, the camera models are randomly moved and rotated.

For the real sets they are initialized from an approximate model based on the CAD of the device. Intrinsic camera parameters, including radial distortion parameters, have been estimated for each camera using the methods described in [12].

To analyze the results, the final global error mean, $g_e$, or the final reciprocal error mean, $r_e$, can be considered. Nevertheless those metrics are directly optimized by the algorithm, thus more independent metrics are more appropriate to test the accuracy of the process.

For the virtual calibration sets, the calibration model is known. Hence, a metric comparing geometrically the camera positions of this model and the model obtained by the algorithm can be calculated. In this work, the mean of the error distances between the camera positions is employed.

Another interesting metric is the dispersion of the sphere center distance of each captured token of a calibration set. This value is straightforward to compute because the algorithm in each iteration estimates the 3D position of the sphere centers of each captured token. In this work, the dispersion is computed as the difference between the minimum and the maximum of this value for a calibration set. Ideally, the sphere center distance average of the captured tokens of a calibration set should match the token reference distance with null dispersion.

Figures 12, 13 and 14 shown an example of evolution of the algorithm in each case. The line labeled as *Reference* in the figures represents the evolution of the average of the sphere center distances in the calibration set. As can be seen in Figure 12, using the exact sphere center projections, the algorithm attains null global error mean, null reciprocal error mean and exact scale with only a scale correction. On the other hand, using the estimated sphere center projections, the global and reciprocal error means cannot attain a null value, and several scale corrections are performed until convergence (Figure 13). Each gap in the reciprocal error mean represents a scale correction step in the algorithm. Finally, for real calibration sets, the convergence becomes more difficult (Figure 14).

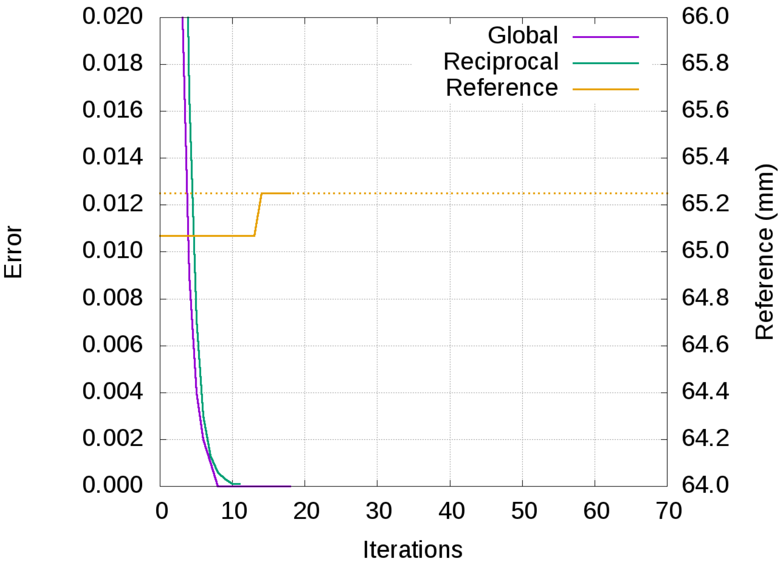

Figure 12: Calibration evolution: virtual calibration set and exact sphere center projections. The global and reciprocal error mean, and the sphere center distance mean (*Reference*) are shown. The dotted line represents the goal reference distance.

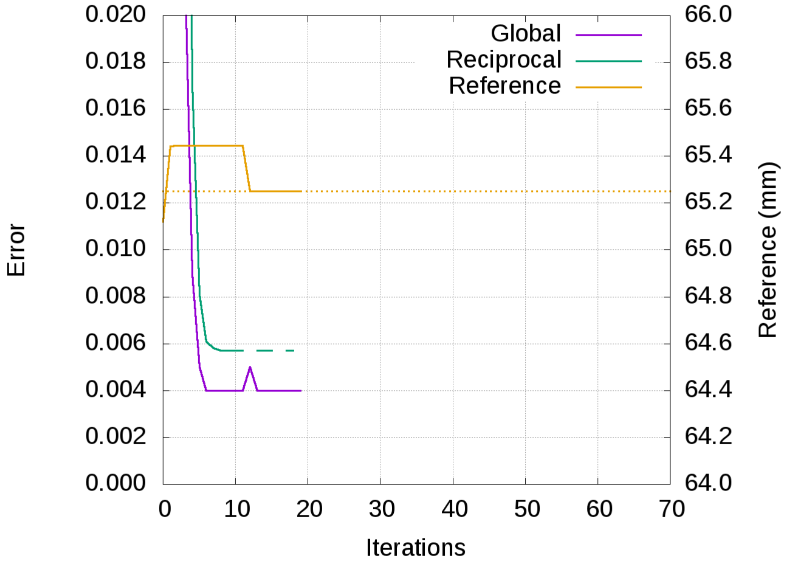

Figure 13: Calibration evolution: virtual calibration set and estimated sphere center projections.

In the real sets, several noise sources appear, hence the algorithm needs more iterations to converge. Among the noise sources, the following can be considered: inaccuracies in the intrinsic parameters (radial distortion, ...), tiny delays between the cameras during the synchronized capture because the token is freefalling, additional errors in sphere center projection estimation due to digital noise on images or uneven illumination, ...

Tables 2 and 3 present statistics summarizing the results obtained for the different calibration sets. Table 2 shows the dispersion of the token sphere distances. For each experimental setup, five runs of the algorithm have been executed, one with each calibration set. The mean, the standard deviation and the minimum and maximum values are presented. As expected, there is no dispersion for the virtual sets using exact center projections. Nevertheless while using the estimated center projections in both the virtual and real setups a minimal dispersion appears. The real setup obtains a slightly higher dispersion due to other noise sources as previously commented.

On the other hand, Table 3 provides the statistics of the geometrical comparison between the original and the obtained calibration. Again, the algorithm with the virtual setup and the exact center projections attains the

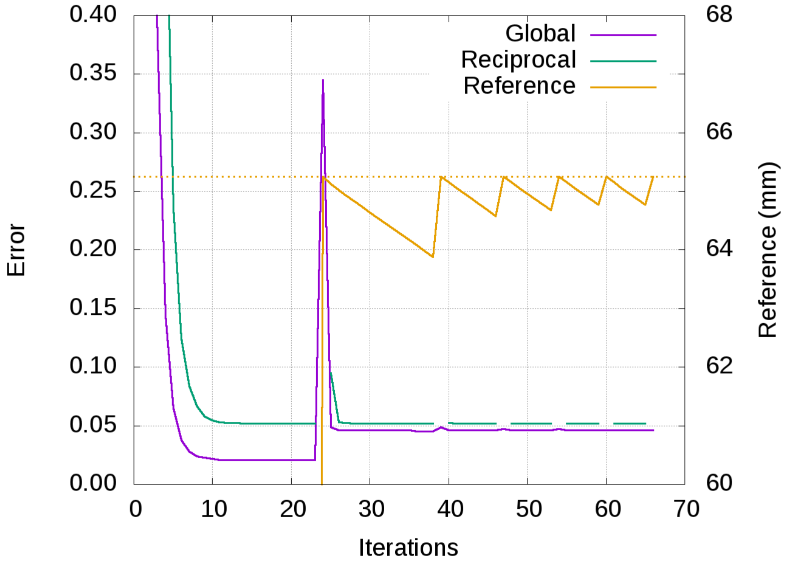

Figure 14: Calibration evolution: real calibration set.

exact calibration, while the setup using the estimations gets a really good approximation.

Table 2: Sphere center distance dispersion statistics for the different experiments. Units are in millimeters.

| **Sets** | **Centers** | **Mean** | **Std. Dev.** | **Min.** | **Max.** |
|---|---|---|---|---|---|
| Virtual | *exact* | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | *estimated* | 0.0284 | 0.0029 | 0.0240 | 0.0330 |
| Real | *estimated* | 0.0358 | 0.0070 | 0.0270 | 0.0470 |

Table 3: Geometric calibration error statistics for the different experiments. Units are in millimeters.

| **Sets** | **Centers** | **Mean** | **Desv.** | **Minimum** | **Maximum** |
|---|---|---|---|---|---|
| Virtual | *exact* | 0.00009 | 0.00004 | 0.00006 | 0.00017 |
| | *estimated* | 0.39017 | 0.01245 | 0.37285 | 0.40776 |

Finally, using the methodology in [4] a 3D reconstruction of the token can be made using a calibration and a capture set. Figure 15 shows a pair of reconstructions using an accurate and a inaccurate calibration. In the case of the inaccurate calibration the reconstruction is poor.

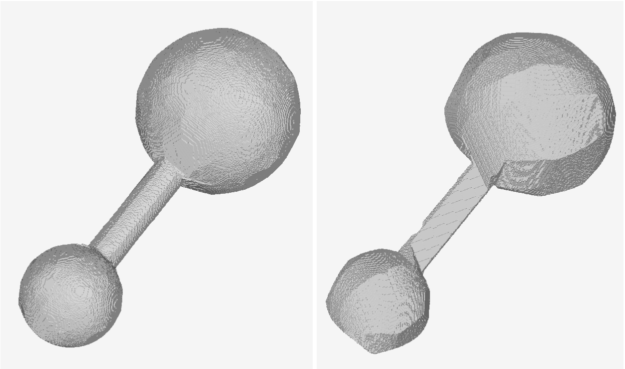

Figure 15: Token reconstruction using a accurate (left) and a inaccurate (right) calibration.

## 5. Conclusion

The aim of this work was to present a method to simplify the calibration of the extrinsic parameters of a device in an environment using multiple cameras, with a high precision requirements. The method proposes the use of a simple calibration token composed of two spheres of different sizes that are connected by a slim rod. The token is made of a solid color, and hence is simple to build. Only an exact knowledge of the distance between the sphere centers is required.

The presented method modifies an existing calibration algorithm that used single spheres as calibration tokens, avoiding the scale calibration problem inherent to that algorithm. The result is a more precise estimation of the extrinsic parameters that increases the performance results of the techniques using this calibration.

Additionally, the use of the new token can reduce the number of captures needed for the algorithm to converge adequately. On the one hand the new token is composed of two spheres, hence the double of calibration points are available per capture if occlusions are not taken into account. And on the other hand, if the token is presented to the system in motion, the token *flies* worse than a single sphere presenting more variate views to the cameras. This behavior accelerates the algorithm convergence.

Although the experiments were performed with a specific camera setup (in this case to obtain 3D representation of objects) the method can be employed in other multi-camera scenarios. For example, in a production line where robots or other automations have to insert or install a part in a product (assembling a toy, placing the windshield in a car, crimping a gem in a jewel, ...) a set of few cameras can be installed around the assembling area and can be calibrated using the method. An good calibration allows to locate the product and to determine its orientation accurately, thereby the assembly can be controlled on real time.

## References

[1] O. Faugeras, "Three-dimensional computer vision: a geometric viewpoint," MIT Press, 1993, ISBN: 0-262-06158-9.

[2] R. I. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision," 2nd Ed. Cambridge University Press, 2004, ISBN: 0521540518.

[3] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,", *IEEE J. of Robotics and Automation*, Volume 3, Issue 4, 1987, pp. 323–344.

[4] J.C. Perez-Cortes, A.J. Perez, S. Saez-Barona, J.L Guardiola and I. Salvador, "A System for In-Line 3D Inspection without Hidden Surfaces," *Sensors*, 2018, 18, 2993.

[5] L. Gracia, S. Saez-Barona, D. Carrion, I. Salvador and J. Perez-Cortes, "A System for Real-Time Multi-View 3D Reconstruction," *2010 Workshops on Database and Expert Systems Applications*, 2010, pp 235–239.

[6] Z. Bi and L. Wang, "Advances in 3D data acquisition and processing for industrial applications, "*Robot. Comput. Integr. Manuf.*, 2010, 26, pp 403–413.

[7] C.R. Dye, "Volumetric Scene Reconstruction From Multiple Views", *Foundations of Image Understanding*, Ed. Springer: New York, NY, USA, 2001, Volume 628, pp. 469–489.

[8] R. Penne, B. Ribbens and P. Roios, "An Exact Robust Method to Localize a Known Sphere by Means of One Image," *International Journal of Computer Vision*, 2019, Issue 127, pp 1012–1024.

[9] W. Andrew W, R.B. Fitzgibbon and A. Fisher. "Buyer's Guide to Conic Fitting," *Proc. 5th British Machine Vision Conference*, Birmingham, 1995, pp. 513–522.

[10] R.C. Gonzalez and R.E. Woods. "Digital Image Processing," Ed. Pearson Education, 2011.

[11] K. Fukunaga. "Introduction to Statistical Pattern Recognition," Ed. Elsevier, 1990.

[12] Z. Zhang. "A Flexible New Technique for Camera Calibration," *IEEE Transactionss on Pattern Analysis and Machine Intelligence*, 2000, Volume 22, pp. 1330–1334.