MONOCULAR CAMERA CALIBRATION USING PROJECTIVE INVARIANTS

Vilca Vargas Jose R[1] and Quio Añauro Paúl A[2] and Loaiza Fernández Manuel E[3]

Universidad Católica San Pablo, Arequipa Perú

ABSTRACT

Camera calibration is a crucial step to improve the accuracy of the images captured by optical devices. In this paper, we take advantage of projective geometry properties to select frames with quality control points in the data acquisition stage and, further on, perform an accurate camera calibration. The proposed method consists of four steps. Firstly, we select acceptable frames based on the position of the control points, later on we use projective invariants properties to find the optimal control points to perform an initial camera calibration using the camera calibration algorithm implemented in OpenCV. Finally, we perform an iterative process of control point refinement, projective invariants properties check and recalibration; until the results of the calibrations converge to a minimum defined threshold.

KEYWORDS

Camera calibration, Pattern recognition, Optimization & Frontoparallel projection.

1. INTRODUCTION

The main objective of computer vision area can be defined as the creation of systems that, through the analysis and biological inspired image processing, are able to "understand" a visual input [1]. To achieve this objective, object detection and tracking is one of the most important processes, because it lets the computer get a better model of the real world. Optical technology is widely used because of its low costs and availability. Also, optical technology has the great advantage that the tracking object does not have to be overposed by cables or other components [2].

Camera calibration is an essential step in computer vision, image processing and optical measuring. Since the accuracy at computing the 3D position of the object is directly related to a precise camera calibration [2,3].

Camera calibration faces some error sources such as the non-linear distortion caused by a non-frontoparallel input image, the imperfection of the calibration object, lens distortion or the lack of certainty when locating the control points directly from the geometric calibration patterns [4,5]. To face these problems, most modern algorithms include an iterative frontoparallel reprojection and pattern refinement step [5].

We propose an optimized camera calibration method based on the use of projective invariants properties to detect invariant projective patterns that grant sturdiness and a better accuracy in the data acquisition stage and take advantage of them in the next steps, alongside the use of the algorithm proposed by Zhang [6] implemented in OpenCV [7] to find the best values of the intrinsic and extrinsic parameters of the camera.

David C. Wyld et al. (Eds): AIAPP, NLPML, DMA, CRIS, SEC, CoSIT, SIGL - 2022
pp. 253-272, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120921

The rest of this paper is organized as follows. In Section 2, we present a brief collection of related works. In Section 3 we detail our proposed method. In Section 4 we present the test environments and the results of the performed experiments. Later in Section 5 we present the accuracy comparison expressed in centimetres, and finally in Section 6 we present our conclusions and future work.

2. RELATED WORK

2.1. Camera Calibration

Camera calibration problem has been extensively studied. According to Song et al. [8] it can be divided in three main categories. Traditional camera calibration, camera auto-calibration and camera calibration based on active vision, although we consider hybrid methods as well. Thus, we consider the hybrid methods as a fourth category.

- **Traditional camera calibration** consists of the calculation of the intrinsic and extrinsic parameters of the camera through mathematical transformations after the processing of the images. It acquires an advantage when the shape and size of the calibration object is known. One of the most visited and well-known methods was proposed by Zhang [6], consisting on capturing several angles of a plain calibration pattern with the camera and restrict the internal and external camera parameters analysing the relation between each control point in the plain pattern and the corresponding control point in the image, and finally, performing a non-linear optimization of the calculated results by a maximum similarity criteria. In 2009, Datta et al. [4] recognized the deficiency on the control points localization as an error source and revisited Zhang's method proposing an iterative control point refinement strategy. We can find many novelty calibration techniques in the last decade, replacing the plain calibration pattern with a hanging chain curve [9], spheres [10] or using the motions of a wand [11]. Liu et al. [12] propose a method that presents a new calibration target which uses projective invariants properties to find feature points and feature lines to reduce the image distortion. Sarmadi et al. [13] propose a method that calculates the extrinsic parameters of multiple cameras and the relative position between the cameras and a rigid set of planar markers at each frame.

- **Camera auto-calibration** does not depend on the reference calibration object. The calibration is performed through the comparison of the relationship between the environment images through the camera movement. Yao et al. [14] proposes a camera auto-calibration method that consists of the construction of a trajectory matrix through the tracing of characteristic points, low range decomposition of the trajectory matrix and sparse restriction, and a sturdy homography matrix estimation. Chen [15] proposes a method of camera auto-calibration based on a geometrical analysis from four coplanar corresponding points and a fifth non-coplanar.

- **Camera calibration based on active vision** consists of acquiring several images after controlling the camera to perform a defined special movement. Afterwards, the intrinsic and extrinsic parameters can be obtained linearly using both the acquired images and the known movement trajectory. De Ma Sang [16] proposes a method of camera calibration based on triorthogonal translation movement that can calculate both intrinsic and extrinsic parameters through three to six translational movements.

- **Hybrid methods** combine more than one camera of the aforementioned categories. Loaiza et al. [2] proposes a hybrid method for stereo camera calibration using both a traditional photogrammetric method to calibrate each camera, and an auto-calibration method to compute the extrinsic parameters of the cameras related to the position between them.

2.2. Projective invariants

The transformation of 3D coordinates in the real world to 2D coordinates in the image is known as camera projection. Anyway, there is a different need to calculate the rules to transform the coordinates from a point in an image to the coordinates in another one, as well known as 2D homography [17].

This transformation must be able to calculate the coordinates on an image to an infinitely distant point; for which, Euclidean geometry is not powerful enough. However, those cases can be handled by projective geometry.

In projective geometry, parallelism and orthogonality are not necessarily present, and the distance between two points can be affected in the coordinates transformation, as seen in Figure 1. Nevertheless, there are certain properties that are not affected in projective transformations, as listed by Clemens [18], called projective invariants.

1. **Collinearity and coplanarity:** A defined set of points is considered collinear if every point rest on a same line (see Figure 1). In this way, any pair of point in R^2 will be always collinear.

   The collinearity of three points on a two-dimensional space $v_1, v_2, v_3 \in R^2$ can be defined as

   $$\begin{vmatrix} v_1 & v_2 & v_3 \\ 1 & 1 & 1 \end{vmatrix} = 0, \qquad (1)$$

   where, if the determinant is different to zero, the points are not collinear. Similarly, each set of n points that belong to R^m are collinear if the distance from points $v_3, v_4, \ldots, v_n$ to the line defined by points v_1 *and* v_2 is, for each point, equal to zero.

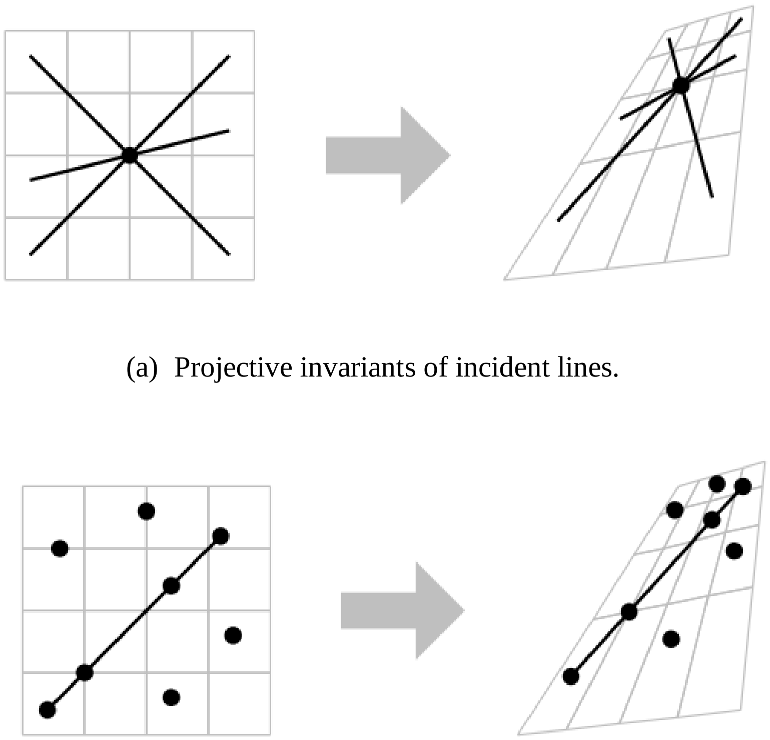
   A set of points is considered coplanar if there is a plane that contains all the set of points. In this way, any set of three points in R^3 is always coplanar.

   The coplanarity of four points in a three-dimensional space $w_1, w_2, w_3, w_4 \in R^3$ can be defined as

   $$\begin{vmatrix} w_1 & w_2 & w_3 & w_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0, \qquad (2)$$

   where, if the determinant is different to zero, the points are not coplanar. In this way, a set of n points that belong to R^m are coplanar if the distance of the points $w_4, w_5, \ldots, w_n$ to the plane defined by the points w_1, w_2 *and* w_3 is, for each point, equal to zero.

   After any projective transformation, the collinear points stay collinear, and coplanar points stay coplanar.



(a) Projective invariants of incident lines.

(b) Projective transformation example with collinear and coplanar points.

Figure 1. Projective invariants transformations, extracted from Clemens [18].

2. **Incidence:** The incidence criteria means that three or more lines are concurrent, that is to say, they intersect each other in the same point (see Figure 1a).

   Thus, for three lines described as the equations

$$\begin{array}{ll} a_1x + b_1y + c_1 & = 0 \\ a_2x + b_2y + c_2 & = 0 \ , \\ a_3x + b_3y + c_3 & = 0 \end{array} \tag{3}$$

are concurrent if

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} = 0, \tag{4}$$

where, if the determinant is not equal to zero, the lines are not concurrent.

In the same way, a set of *n* lines in R^m, each one defined by two points *(v_{11}, v_{12}), ..., (v_{n1}, v_{n2})* are concurrent if every possible combination of lines has the same intersection point.

3. **Cross relations:** A cross relation is the product of two relations. Hence, each cross relation needs four different values. Cross relations are invariant after any projective transformation.

   - Collinear points:

     The cross relation of the distance of four collinear points is defined as:

$$CR(A,B,C,D) = \frac{\overline{AC}}{\overline{CD}} \cdot \frac{\overline{BD}}{\overline{AD}}, \tag{5}$$

     where $\overline{XY}$ represents the Euclidian distance between point *X* and point *Y*. The points are ordered alphabetically from *A* to *D* as seen in Figure 2a.

     In this way, the cross relation of collinear points is given when we can trace a straight line that connects four points *A*, *B*, *C* and *D* so the distance between points can be replaced in Equation 5.

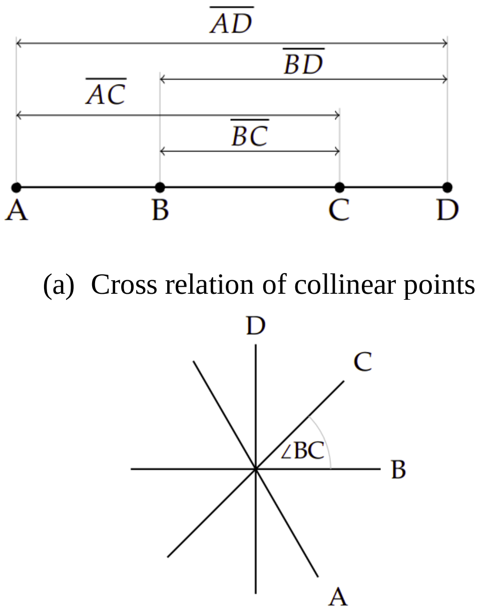
   - Concurrent lines:

     The cross relation of the angles of four concurrent lines is defined as:

$$CR(A,B,C,D) = \frac{\sin(\measuredangle AC)}{\sin(\measuredangle CD)} \cdot \frac{\sin(\measuredangle BD)}{\sin(\measuredangle AD)}, \tag{6}$$

     where $\measuredangle XY$ represents the angle formed by the line *X* and the line *Y*. Lines are ordered alphabetically from *A* to *D* counterclockwise as seen in Figure 2b.

     In this way, the cross relation of collinear points is given when a set of four lines *A*, *B*, *C* and *D* collide in the same point. Therefore, the Equation 6 must be replaced with the angles formed between each pair of lines.



(a) Cross relation of collinear points.

(b) Cross relation of concurrent lines.

Figure 2. Cross relationships, extracted from Clemens [18].

3. PROPOSED METHOD

Our proposed method is based on an iterative control points refinement process through a frontoparallel projection and reprojection, in addition to a well distributed location of the pattern calibration control points across the size of the frame and the usage of projective invariants properties to assure the data acquisition of quality calibration control points.

We use a calibration pattern that consists of 20 concentric rings divided in 5 columns and 4 rows. The concentric rings pattern show a great performance due to the presence of two centroids that, in an ideal situation, are the same point [4]. The number of columns and rows correspond as the minimum rings quantity to recognize and validate the projective invariants properties of collinearity and angle cross relations, as well as the minimum quantity to achieve a rectangular pattern, so its orientation can be defined by the position of rows and columns in the captured image.

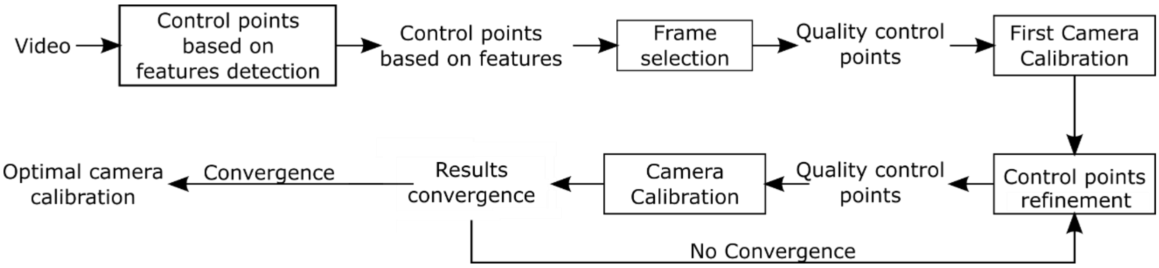


Figure 3. Pipeline of the processes performed in the proposed method.

As seen in our pipeline (Figure 3), our method is based in four principal steps:

1. **Control points based on feature detection** where we locate the position of control points in the input image based on the features of the calibration pattern used.

2. **Frame selection** where we use a grid to distribute the calibration patterns through the frame size and we evaluate the projective invariants properties in order to get a set of well distributed frames which control points satisfy invariant projective properties of collinearity and cross-angle relationship.

3. **First camera calibration** where we perform a first estimation of the camera matrix using OpenCV [7] algorithm.

4. **Control points refinement** where we look to improve the quality of the collected control points performing a frontoparallel projection and a reprojection to avoid distortions and update the correspondent control points.

5. **Camera calibration and results convergence** where we perform the camera calibration to achieve the next estimation of the camera matrix until a convergence point is reached.

3.1. Control points detection based on features detection

The control points detection step consists in four sub steps.

First, we mask the input image to cover just the minimum area containing all the calibration pattern.

Then, we apply an adaptive threshold to divide the background (the white surface of the pattern) from the foreground (the rings).

After acquiring our segmented input, we search for all the isolated ellipses using the OpenCV function *findContours*. We use a contour hierarchy where we must find an exterior contour (father) and an interior contour (son).

Finally, once we have all the ellipses in the input, we define our pattern rings and their centers. To achieve this, we must have some restraints for each ellipse.

- Each ellipse must have at least two near ellipses in a radius not longer than five times the radius of the ellipse from which the comparison is being generated.
- If we have more than 20 ellipses that fulfil the aforementioned father-son hierarchy defined in the previous sub step, we should discard ellipses with different fathers.

To define the centre of the ellipses we use the OpenCV method *fitEllipse* to get an approximation of each ring centre and then we calculate the ellipse getting the average centre of both rings.

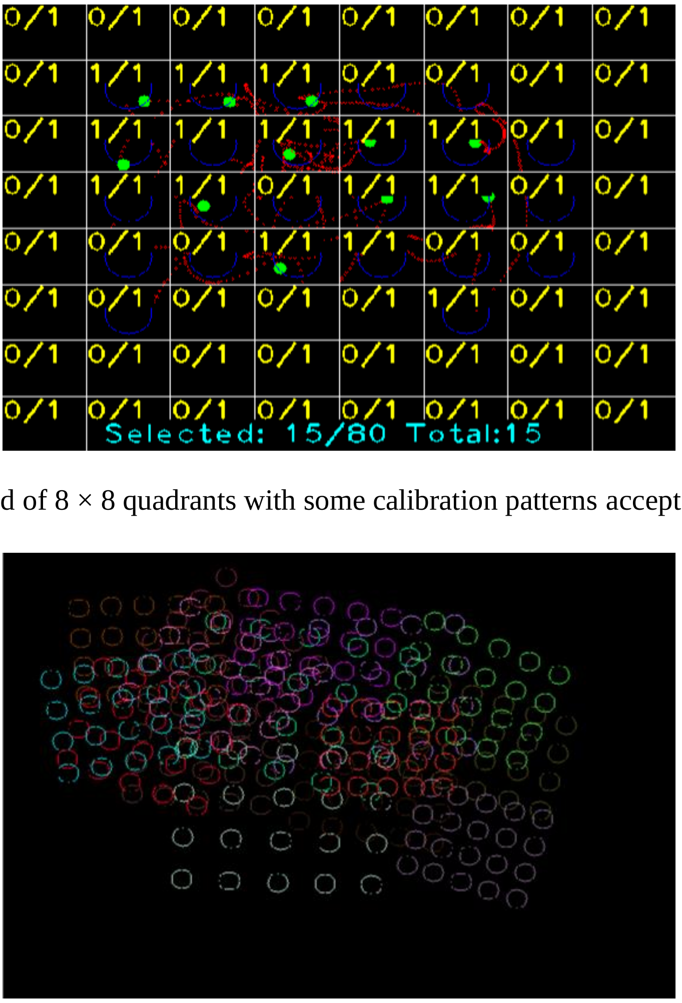
After performing this sub steps, we must have a total of 20 control points defined by the average centre of the concentric rings. In case the total of control points is less than 20, the frame is not considered as a candidate for the next steps.

3.2. Frame selection

To get a precise calibration we must find frames where the control points are as well distributed as possible around the frame size. To achieve that, we define a grid which cells must contain the centre of the calibration pattern inside each acceptation area to define the frame as a potential frame.

In Figure 4 we can see an example of an *8 x 8* grid where 15 frames had been already selected. As the grid has been defined, each cell can contain a maximum of one potential frame. Thanks to this, the 15 selected potential frames are distributed across a wider section of the frame size. Otherwise, two potential problems would arise. In first instance, the selected potential frames would be agglomerated in small areas of the frame size. The second problem is that we would have to request a high quantity of potential frames, since we would accept every potential frame from the aforementioned agglomerations.

Once we define a potential frame based on its position along the frame size, we evaluate the projective invariants properties of its calibration pattern. We evaluate the collinearity and the cross-angle relationship of the control points of the pattern.



0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
0/1	1/1	1/1	1/1	0/1	0/1	0/1	0/1
0/1	1/1	1/1	1/1	1/1	1/1	0/1	0/1
0/1	1/1	1/1	0/1	1/1	1/1	0/1	0/1
0/1	0/1	0/1	1/1	1/1	0/1	0/1	0/1
0/1	0/1	0/1	0/1	0/1	1/1	0/1	0/1
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

Selected: 15/80 Total:15

(a) Grid of 8 × 8 quadrants with some calibration patterns accepted.

(b) Control points distribution from the calibration pattern in the selected frames.

Figure 4. Frame selection process.

Collinearity: We evaluate the collinearity of the points that belong to each line as seen in Figure 5, where we can see the set of collinear points matched by red, yellow, blue and green lines for the vertical, horizontal and both diagonals respectively. Also, we are able to see black lines that represent the set of points that will not be evaluated because they will always be collinear.

In this way, we define 17 sets of collinear points distributed in four possible orientations:

- **Horizontal collinear points:**
    1. *point [1], point [2], point [3], point [4] & point [5]*
    2. *point [6], point [7], point [8], point [9] & point [10]*
    3. *point [11], point [12], point [13], point [14] & point [15]*
    4. *point [16], point [17], point [18], point [19] & point [20]*
- **Vertical collinear points:**
    5. *point [1], point [6], point [11] & point [16]*
    6. *point [2], point [7], point [12] & point [17]*
    7. *point [3], point [8], point [13] & point [18]*
    8. *point [4], point [9], point [14] & point [19]*
    9. *point [5], point [10], point [15] & point [20]*

- **Top left to bottom right collinear points:**
  10. *point [11], point [7] & point [3]*
  11. *point [16], point [12], point [8] & point [4]*
  12. *point [17], point [14], point [9] & point [5]*
  13. *point [18], point [14] & point [10]*
- **Bottom left to top right collinear points:**
  14. *point [6], point [12] & point [18]*
  15. *point [1], point [7], point [13] & point [19]*
  16. *point [2], point [8], point [14] & point [20]*
  17. *point [3], point [9] & point [15]*

We trace the best line approach between all the supposedly collinear set of points. Then, we calculate the Euclidean distance between each of these points and the traced line. If this distance is less or equal to our threshold then it is accepted as a collinear point. In case one or more of the set of points are not collinear, the frame is discarded as a potential calibration frame.

In Figure 7a we can see an accepted frame with collinear control points represented by blue lines that match all the defined sets of collinear control points.

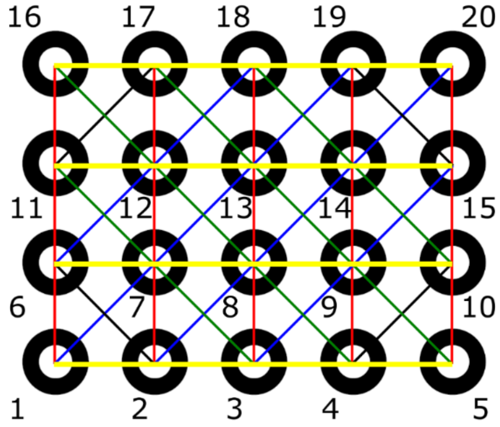


Figure 5. Collinear points in the calibration pattern.

Cross-angle relationship: Once the collinearity is successfully evaluated, we proceed to evaluate the cross-angle relationship of the inner quadrant of the pattern (*point[6], point[7], point[8], point[11], point[12], point[13]*).

We calculate the cross-angle relationship of each point as

$$\frac{\sin(90°)}{\sin(45°)} \cdot \frac{\sin(90°)}{\sin(135°)} \cong \frac{\sin(\measuredangle AC_{ij})}{\sin(\measuredangle CD_{ij})} \cdot \frac{\sin(\measuredangle BD_{ij})}{\sin(\measuredangle AD_{ij})} \tag{7}$$

where *i* is the frame being evaluated and *j* is the evaluated point. The angles are formed as seen in Figure 6, where *A*, *B*, *C* and *D* are enumerated counterclockwise represented by a red, yellow, green and blue line respectively. If the difference of the calculated angle relationship is greater than our threshold, then the frame is discarded as a potential calibration frame.

In Figure 7b we can see an accepted frame with the respective cross-angle relationship for each point of the inner quadrant.

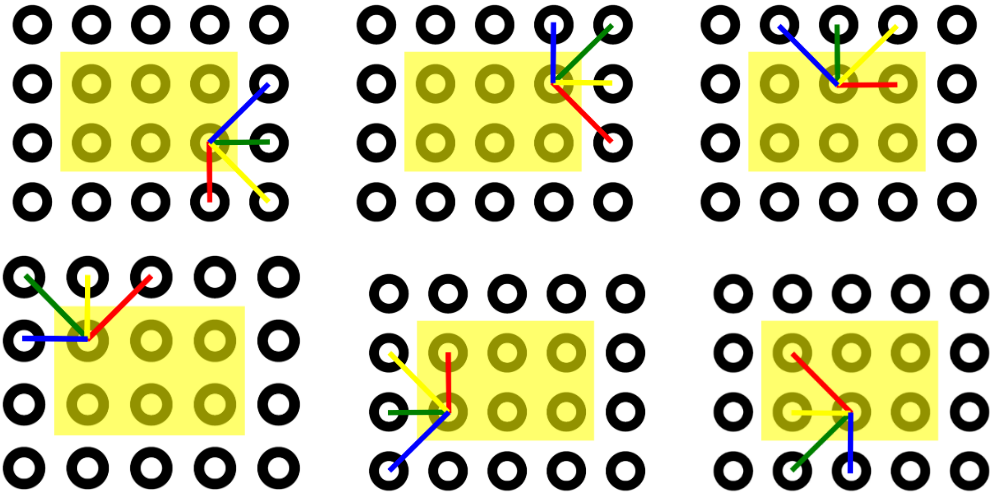
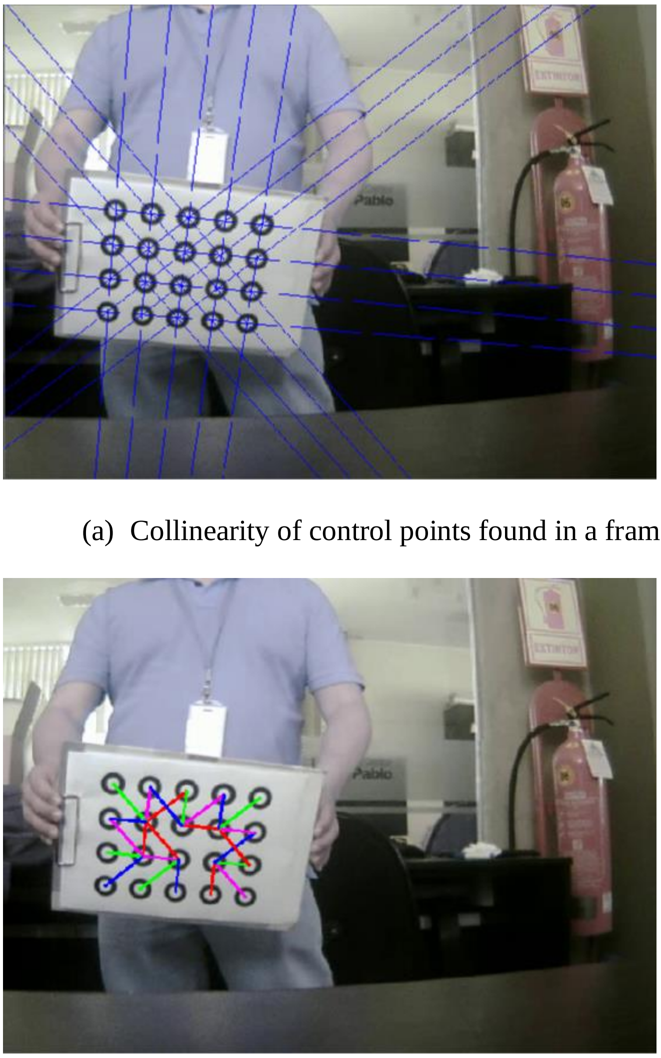


Figure 6. Cross-angle relationship detail of the calibration pattern inner quadrant.



(a) Collinearity of control points found in a frame

(b) Cross-angle relationship of control points found in a frame.

Figure 7. Evaluation of collinearity and cross-angle relationship in a frame.

3.3. First camera calibration

We perform a first camera calibration using all the calibration frames that were not discarded in the previous step. We use the camera calibration algorithm implemented in OpenCV to get an initial estimation of the camera matrix.

This step is crucial and must always be done because the first estimation of the camera matrix is needed to perform the next step. As well, the RMS of this calibration is the base line we look forward to improve in the subsequent steps, as this step performs a non-optimized calibration. In other words, it performs the calibration implemented in OpenCV as detailed by Zhang [6] and it does not take advantage of the control points refinement and their projective invariants properties evaluation as detailed further in this section.

3.4. Control points refinement

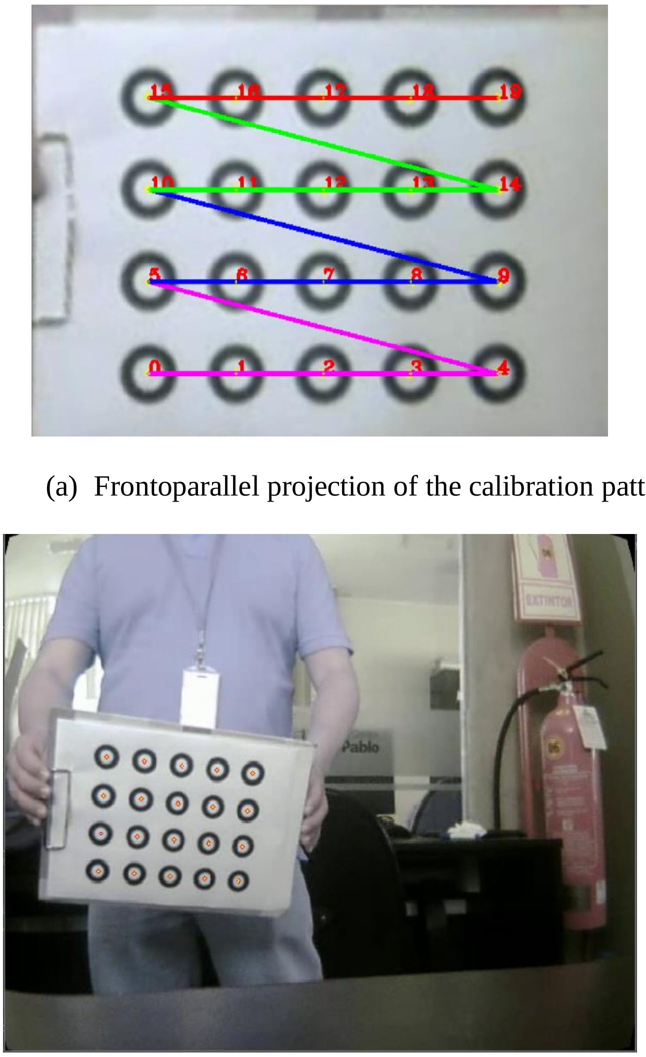
We use the calculated parameters of the first camera calibration to execute a refinement process of the control points. First, we undistort the input image and reproject it in a frontoparallel plane in the world coordinate system as seen in Figure 8a. In this projection we proceed to do a search for the control points. This search must be faster since the image has been undistorted and unprojected, therefore the location and orientation of the calibration pattern grants us a better distinction of where the control points must be located.

Once we find all the control points in the frontoparallel projection, we proceed to reproject them in the original image using the first camera calibration calculated parameters into the camera coordinate system. We use the set of control points found in the original image (old points) as well as the reprojected set of control points (new points) to update the new set of points:

1. For each row we fit a line using the respective points from the new set of points.
2. For each point in the row
   a. Calculate the distance of the original control point to the line.
   b. Calculate the distance from the reprojected point to the line.
   c. Calculate the blend factor as the proportion of each distance to the line giving a priority to the closest one, where the factor is less or equal than 1.
   d. Update the control point using the blend factor

$$newPoint = newPoint * factor + oldPoint * (1 - factor) \tag{8}$$

Once we update our set of control points as seen in Figure 8b, we proceed with the projective invariants properties evaluation, as defined in the second step of the process. If the updated control points do not satisfy the projective invariants properties, then the frame is discarded.



(a) Frontoparallel projection of the calibration pattern.

(b) Control points reprojection in original image.

Figure 8. Control points refinement process

3.5. Results convergence

Once the refinement process is completed, we perform another camera calibration using the same algorithm implemented in OpenCV.

The obtained RMS is then compared to the previous one (the RMS from the first iteration is compared with the RMS from the first camera calibration). If the absolute value of our RMS difference is greater than our convergence threshold, then the obtained camera matrix and RMS becomes the new data to perform the control points refinement process into the next iteration.

Otherwise, if the absolute value of the RMS difference is less or equal to the defined threshold, then we successfully achieved an optimal camera calibration. We can see an example of this process in Table 1 and the graphic results on Figure 9, where our convergence threshold is defined as 0.00005. We can see how, as we iterate, the absolute value of the RMS difference keeps being greater than our threshold. However, when we reach iteration number eight, we get an absolute value of the RMS difference minor to our threshold, consequently we take the calibration of this iteration as optimal.

Table 1. Iterative process and results convergence on PS3 Eye Camera.

# Iteration	**Fx**	**Fy**	**Uo**	**Vo**	**RMS**	**Convergence Value**
First Calibration	716.319	722.284	325.828	244.346	0.33587	-
1	696.936	703.097	330.429	246.309	0.262235	0.073635
2	673.073	680.266	331.41	251.325	0.251114	0.011121
3	670.564	677.487	331.131	257.299	0.23994	0.011174
4	692.207	698.407	334.173	262.209	0.232683	0.007257
5	698.262	704.455	335.011	262.948	0.227108	0.005575
6	699.578	705.77	335.239	263.433	0.227299	0.000191
7	698.871	705.075	335.145	263.302	0.227081	0.000218
8	**699.119**	**705.294**	**335.176**	**263.252**	**0.227034**	**0.000047**

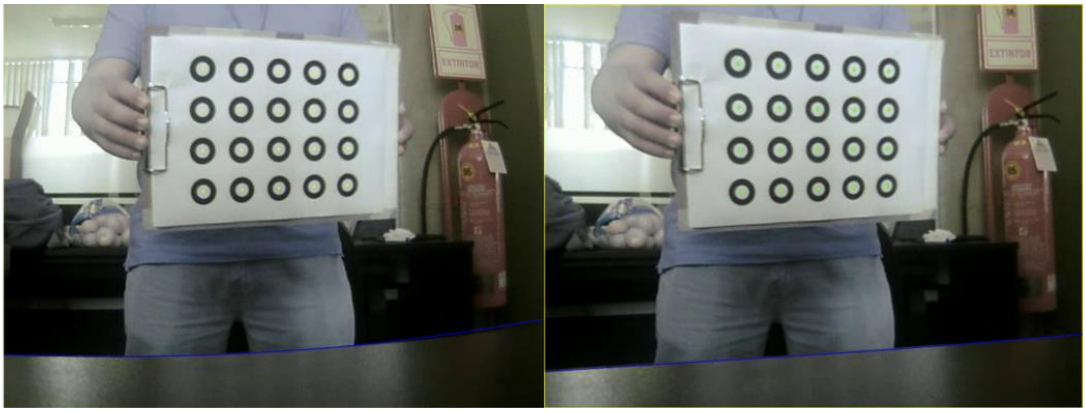


Figure 9. Captured frame from PS3 Eye camera before performing calibration (on the left) and after performing the proposed method (on the right).

4. TEST ENVIRONMENT AND RESULTS

The purpose of the performed experiments was to achieve the best possible configuration of the proposed method as well as demonstrate the improvement granted by the use of projective invariants properties in our method in comparison with a widely used method like OpenCV camera calibration.

For the acquisition of results, we ran multiple experiments in multiple devices. We specify the four camera devices used as well as the three sets of experiments performed with the proposed method in the following sections.

4.1. Test equipment

To perform the tests, four webcams were used:

- *HP Truevision* integrated camera of a *HP Envy 15*, with 640 x 480 px resolution and 30 FPS image rendering.
- *Genius FaceCam 1000X* analog camera, with 640 x 480 px resolution and 30 FPS image rendering.
- *PS3 Eye Camera*, with 640 x 480 px resolution and 60 FPS image rendering.
- *Logitech Brio 4k Pro Webcam*, with 640 x 480 px resolution and 24 FPS image rendering.

4.2. Test methodology

We performed three experiment sets consisting in the algorithm selection, the relative distance of the calibration pattern to the pinhole camera and the performance of the best selected algorithm.

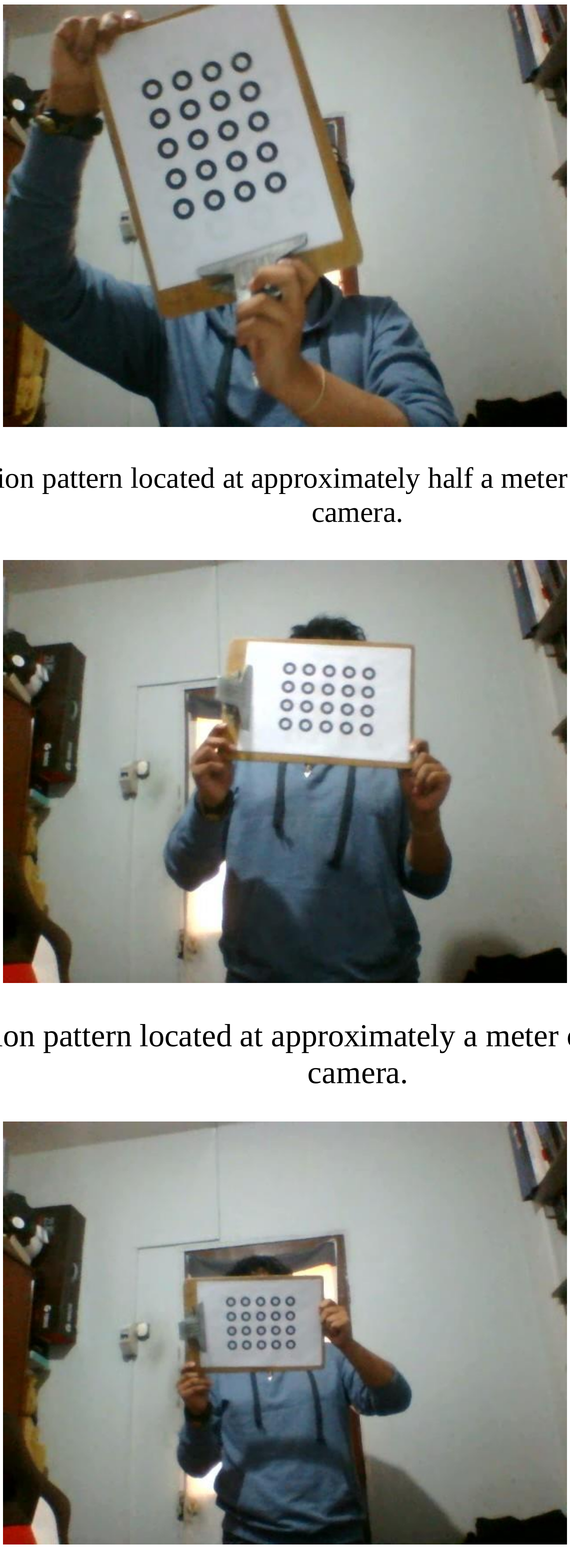
To perform all the experiments, we used a threshold of 0.6 px for the collinearity error, 0.06 for the cross-angle relationship error and we defined a 0.00005 convergence threshold to stop the iterative process considering the quality of the cameras used to perform the experiments which would not allow us to constraint the errors to a lesser value. In the same way, we define the initial size of the grid as *8 x 8* due to the size of the frame, given that the grid cells are small enough to contain the centre of the calibration point at a long distance as far as the calibration pattern is still recognizable for the algorithm considering the quality of the cameras.

1. **Algorithm selection:** We performed an OpenCV calibration compared to four versions of the proposed algorithm to define which one grants better results.

    - **Iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern as proposed by Datta et al. [4] and Prakash et al. [19].
    - **Collinear iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points collinearity.
    - **Cross-angle relationship iterative:** OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points cross-angle relationship.
    - **Iterative with projective invariants properties**: OpenCV camera calibration with iterative refinement of the calibration pattern and evaluation of the control points collinearity and cross-angle relationship.

    To perform these experiments, we require a minimum set of 50 frames. We perform this experiment on three cameras (*HP Truevision*, *PS3 Eye Camera*, *Genius FaceCam 1000x)* with a limit of 10 iterations. Since the number of iterations is low, the minimum set of frames does not need to be high considering that not many frames are going to be discarded in the iterative process.

2. **Relative distance of the calibration pattern:** We perform the proposed method on four different recording of the same camera (*HP Truevision*) where we positioned the calibration pattern at approximately half a meter (Figure 10a), a meter (Figure 10b), meter and a half (Figure 10c) and varied distances identified as close, medium, far and varied respectively to determine which is the best relative distance of the calibration pattern to the camera to perform a camera calibration. To perform this experiment, we require a minimum set of 80 frames since the number of discarded frames is higher due to the verification of projective invariants properties.



(a) Calibration pattern located at approximately half a meter of distance from the camera.

(b) Calibration pattern located at approximately a meter of distance from the camera.

(c) Calibration pattern located at approximately a meter and a half of distance from the camera.

Figure 10. Relative distances of the calibration pattern to the camera

3. **Performance of the best selected algorithm:** We test the performance of the best algorithm from the first experiment combined with the distance of the second experiment to get the general performance of the best selected algorithm. To perform this experiment, we require a minimum set of 80 frames, due to the frames discarded by the projective invariants properties check.

4.3. Results

After performing the explained experiments, we achieve the results detailed in Table 2, Table 3 and Table 4. The results of the first and second experiment (algorithm selection and relative distance of the calibration pattern) affect directly in the last experiment, where we perform the tests with the selected algorithm of the first experiment, and we position the calibration pattern according to the results of the second experiment.

1. **Algorithm selection.** As seen in Figure 11, each version of the proposed method represents a significant improvement in comparison with the OpenCV method which is widely used nowadays. Also, we can see in Table 2 that the improvement percentage is always better when projective invariants properties are applied.

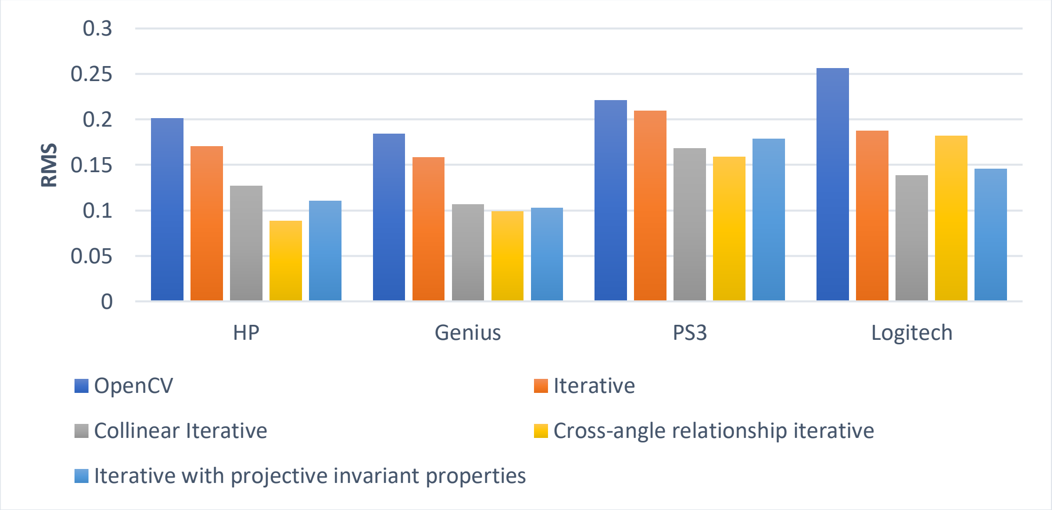


Figure 11. Minimum RMS by algorithm by camera

Table 2. Best improvement percentage by algorithm by camera.

		HP Truevision	**Genius Facecam 1000x**	**PS3 Eye Camera**	**Logitech Brio 4k Pro Cam**
OpenCV	RMS	0.201465	0.183917	0.221083	0.256287
Iterative [4,19]	RMS	0.170335	0.158278	0.209313	0.187268
	% Improvement	15.45	13.94	5.32	26.93
Collinear Iterative	RMS	0.12677	0.106606	0.168134	**0.13837**
	% Improvement	37.08	42.04	23.95	**46.01**
Cross-Angel Relationship Iterative	RMS	**0.088209**	**0.0987823**	**0.159093**	0.181949
	% Improvement	**56.22**	**46.29**	**28.04**	29.01
Iterative with projective invariants properties	RMS	0.110443	0.102717	0.178917	0.145617
	% Improvement	45.18	44.15	19.07	43.18

2. **Relative distance of the calibration pattern.** We can see in Figure 12 how the camera calibration has a better performance when the calibration pattern gets further away from the camera. However, the best performance is reached when the distance is varied containing frames captured with the calibration pattern at diverse distances to the camera. In Table 3 we can notice that the number of iterations grows as we get the calibration pattern further away from the camera. Nevertheless, we get an average quantity when the distance is varied.

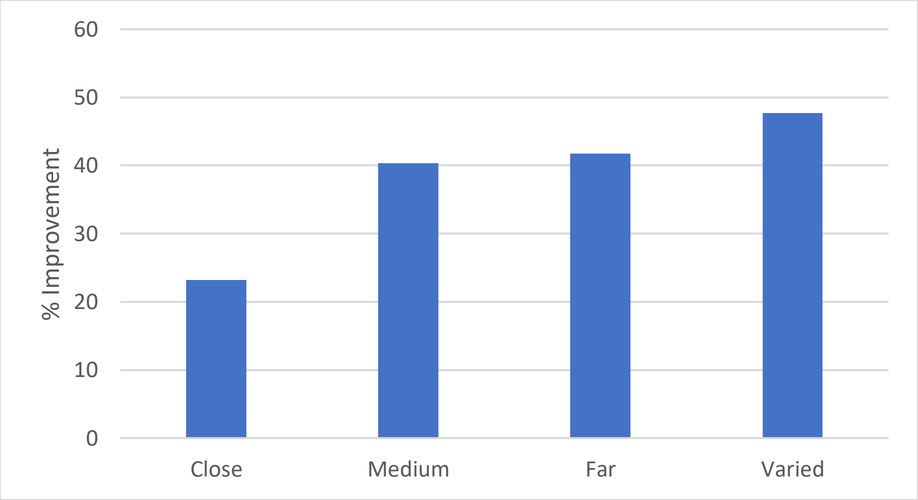


Figure 11. Minimum RMS by algorithm by camera

Table 3. Number of iterations and improvement percentage by the relative distance of the calibration pattern to the camera.

	Close (Half a meter)	**Medium (A Meter)**	**Far (A meter and a half)**	**Varied**
# It	75	**29**	87	65
RMS – OpenCV	0.310345	0.205439	0.173331	**0.170441**
RMS – Our Method	0.238371	0.122536	0.100963	**0.0890842**
% Improvement	23.19	40.35	41.75	**47.73**

3. **Performance of the best selected algorithm.** In Table 4 we observe the performance of the method proposed in all the aforementioned devices. We include the four recordings of *HP Truevision* cameras as well as the recordings of each other camera at varied distances.

Table 4. Improvement percentage and number of iterations of the best selected algorithm.

	HP Truevision				**Genius Facecam 1000x**	**Logitech Brio**	**PS3 Eye Camera**
	Close	Medium	Far	Varied			
# It	33	111	157	55	159	252	9
RMS – OpenCV	0.313	0.2131	0.1736	0.1717	0.1589	0.2317	0.2139
RMS – Our Method	0.2197	0.1274	0.0962	0.103	0.1083	0.144	0.1787
% Improvement	**29.81**	**40.22**	**44.59**	**40.01**	**31.84**	**37.85**	**16.46**

5. ACCURACY COMPARISON

Finally, this section analyzes the accuracy of our method and compares it with other available methods. To perform the comparisson, we use the same setup in each set of results.

We use AprilTag [20] to detect the distance between camera and the pattern. To perform the tests, we use a setup that consists on the pattern tied to the camera with a nylon thread, to reduce obstruction to the minimum, with a radius of 30.7 cm. Also, the position of the pattern was placed throughout all the frame to detect possible distortions in the borders.

We perform these tests with four different variations:

1. **OpenCV Distorted:** The basic camera calibration using OpenCV without performing an undistortion of the evaluated frames.
2. **OpenCV Undistorted:** The basic camera calibration using OpenCV after performing the undistortion of the evaluated frames with the distortion coefficients provided by the calibration method.
3. **Our Method Distorted:** The camera calibration using our method without performing an undistortion of the evaluated frames.
4. **Our Method Undistorted:** The camera calibration using our method after performing the undistortion of the evaluated frames with the distortion coefficients provided by the calibration method.

We compare our method with the calibration parameters calculated by the calibration toolbox provided by Bouguet [21] which is used as ground trouth in the literature [9,10,11,12,15] in both versions, distorted and undistorted.

As we can see on Table 5, our proposed method is the most accurate, gaining an improvement of 16.32% compared to the OpenCV method and reaching up to an improvement of 18.75% when a frame undistortion is performed.

Table 5. Accuracy comparison of the proposed method.

Method	**Average distance detected**	**Average Error**	**% Improvement**
OpenCV Distorted	30.32345931	0.544874483	0
OpenCV Undistorted	30.38143345	0.497539655	8.69
Our Method Distorted	30.67633564	0.455933455	16.32
Our Method Undistorted	**30.80322836**	**0.442726545**	**18.75**
Bouguet Distorted [21]	31.55786665	0.955424898	-75.35
Bouguet Undistorted [21]	32.07657612	1.45828784	-167.64

6. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the modelling and implementation of a new camera calibration approach. The main novelty is the use of projective invariants properties to grant the quality of control points in data collection. Tests demonstrate that the proposed method brings a great improvement in camera calibration quality compared with one of the most used methods (OpenCV). Also, the sector segmentation brings a robust calibration thanks to the well distributed control points.

The proposed method brings its best performance when the calibration pattern is allocated in varied distances during data collection. In addition, the use of a convergence threshold grants us the certainty to achieve an optimum camera calibration. Although, the number of iterations remains variable in each case.

Additionally, the use of projective invariants properties allows us to discard false positive potential frames in frame selection caused by distortion in the frontoparallel projection and reprojection processes.

In the future, we propose to study the relation between average collinearity and average cross-angle relationship errors and camera distortion so that the static distortion parameters in camera calibration could be selected efficiently and automatically. In addition, we propose to model a new cost function to validate the reprojection error of the pattern centroids. Also, we propose the implementation of techniques that automatically improve the acquisition of quality control points to increase the robustness of the proposed method based on distance of the calibration pattern from the camera or the angle of the calibration pattern.

ACKNOWLEDGEMENTS

We acknowledge the financial support of the “Proyecto Concytec - Banco Mundial”, through its executing unit “Fondo Nacional de Desarrollo Científico, Tecnológico y de Innovación Tecnológica (Fondecyt)”, for his research work entitled “Reconstrucción y modelado 3D de las superficies de componentes y piezas de maquinaria pesada usada en Minería, con nivel de precisión milimétrica, para su aplicación en un nuevo proceso optimizado de manutención especializada”.

REFERENCES

[1] Walters, D. (2003) Computer Vision, p. 431–435. John Wiley and Sons Ltd., GBR.

[2] Loaiza, M.E., Raposo, A.B., Gattass, M. (2011) Multi-camera calibration based on an invariant pattern. Computers & Graphics 35(2), 198 – 207.

[3] Gai, S., Da, F., Fang, X. (2016) A novel camera calibration method based on polar coordinate. PLOS ONE 11(10), 1–18.

[4] Datta, A., Kim, J., Kanade, T. (2009) Accurate camera calibration using iterative refinement of control points. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. pp. 1201–1208.

[5] Vo, M., Wang, Z., Luu, L., Ma, J. (2011) Advanced geometric camera calibration for machine vision. Optical Engineering 50(11), 110503.

[6] Zhang, Z. (2000) A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 1330 – 1334.

[7] Bradski, G., (2000) The OpenCV Library. Dr. Dobb’s Journal of Software Tools.

[8] Song, L., Wu, W., Guo, J., Li, X. (2013) Survey on camera calibration technique. In: 2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics. vol. 2, pp. 389–392.

[9] Arık O. and Yuksel S.E. (2022) Camera Calibration Using Catenary. IEEE Sensors Journal, vol. 22, no. 6, pp. 5962-5968.

[10] Wong K. -Y. K., Zhang G. and Chen Z. (2011) A Stratified Approach for Camera Calibration Using Spheres. IEEE Transactions on Image Processing, vol. 20, no. 2, pp. 305-316.

[11] Fu, Q., Quan, Q. and Cai, K.-Y. (2015), Calibration of multiple fish-eye cameras using a wand. IET Comput. Vis., 9: 378-389.

[12] Liu W., Wu S., Wu X. and Zhao H. (2019) Calibration Method Based on the Image of the Absolute Quadratic Curve. IEEE Access, vol. 7, pp. 29856-29868.

[13] Sarmadi H., Muñoz-Salinas R., Berbís M. A. and Medina-Carnicer R., (2019) Simultaneous Multi-View Camera Pose Estimation and Object Tracking With Squared Planar Markers. IEEE Access, vol. 7, pp. 22927-22940.

[14] Yao, Q., Nonaka, K., Sankoh, H., Naito, S. (2016) Robust moving camera calibration for synthesizing free viewpoint soccer video. In: 2016 IEEE International Conference on Image processing (ICIP). pp. 1185–1189.

[15] Chen H. (2017) Geometry-Based Camera Calibration Using Five-Point Correspondences From a Single Image. IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 12, pp. 2555-2566.

[16] Sang De Ma (1996) A self-calibration technique for active vision systems. IEEE Transactions on Robotics and Automation 12(1), 114–120.

[17] Hartley, R.I., Zisserman, A. (2004) *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[18] Clemens, K. (2016) Object Tracking using Projective Invariants. Ph.D. thesis.

[19] Prakash, C.D., Karam, L.J. (2012) Camera calibration using adaptive segmentation and ellipse fitting for localizing control points. In: 2012 19th IEEE International Conference on Image processing. pp. 341–344

[20] J. Wang and E. Olson. (2016) “AprilTag 2: Efficient and robust fiducial detection” in Proceedings of the IEEE/RSJ International Conferenceon Intelligent Robots and Systems (IROS),

[21] J. Y. Bouguet J. Y. (2004) Camera calibration toolbox for matlab. Available: http://www.vision.caltech.edu/bouguetj/calib doc/index.html.

© 2022 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.