Research Article Vol. 59, No. 13 / 1 May 2020 / Applied Optics D239

applied optics

Self-localization of an uncalibrated camera through invariant properties and coded target location

Mariana Chan-Ley,[1,3] Gustavo Olague,[1,4,*] Gerardo E. Altamirano-Gomez,[1] and Eddie Clemente[2,4]

[1]CICESE Research Center, EvoVisión Laboratory, Ensenada, Carretera Ensenada-Tijuana 3918, Zona Playitas, Ensenada B.C. 22860, Mexico
[2]TecNM, Ensenada Institute of Technology, Ensenada 22780, Mexico
[3]e-mail: mchan@cicese.edu.mx
[4]e-mail: eclemente@ite.edu.mx
*Corresponding author: olague@cicese.mx

Received 11 December 2019; revised 12 March 2020; accepted 27 March 2020; posted 27 March 2020 (Doc. ID 385841); published 21 April 2020

This paper recalls one of the most critical problems for the area of computer vision, the automatic location of a single camera. Today, several robotic devices rely on technologies other than visual information to perform self-localization. An artificial optical system will significantly benefit from knowing its location within a three-dimensional world since this is a crucial step to approach other complex tasks. In this paper, we will show how to compute the position of the camera through an uncalibrated method making use of projective properties, the projection model of the camera, and some reference points. We introduce a simple yet powerful way to detect coded targets in photographic images. Then, we describe an uncalibrated approach used to identify the location of a camera in three-dimensional space. The experiments carried out confirm the validity of our proposal.

https://doi.org/10.1364/AO.385841

1. INTRODUCTION

The main task for robots to move autonomously in the three-dimensional world is to know its location concerning objects around it. Throughout the years, techniques have been developed to calculate the position of robots based mainly on active sensors such as GPS, sonar, laser, and omnidirectional cameras [1–3]. The image-based camera location problem remains a critical factor in vision-based robotics [4]. The open problem of visual computing is to achieve a solution to the quest of providing robotic systems with an artificial vision system; however, most robots come with integrated cameras, usually only one, whose resolution is not the best, and with a simple lens. If we could calculate the location of such cameras, we could know not only where the robot was but how to compute a trajectory in three-dimensional space. Nevertheless, the automatic location of a camera can be a very challenging problem, especially when we want to use only a single camera.

The calibration matrix is part of the techniques commonly used for the auto-location of the optical center. To calibrate the camera, we extract some point locations accurately from a photograph taken on the scene. The projection matrix is calculated using the inner parameters of the camera, which transforms a point in the scene into the projected point in the photograph. This method can achieve very high accuracy but is vulnerable to failures when the camera changes focus [5,6]. If the position of the camera reorientates, a new calibration is needed.

The problem of indoor localization of a mobile platform based on monocular vision and coding images is an open research area. In Ref. [7], the authors describe a system using a set of coding graphics in combination with resection models to obtain robust positioning results with high accuracy. However, their method relies on collinearity equations and calibration of exterior orientation without an accurate estimation of the camera parameters. Li *et al.* [8] carried out an implementation using stereo camera systems. These devices are usually quite expensive and delicate in handling, and their methodology is computationally costly. Also, it is difficult to manage and implement at the moment of working with real-time systems. In this paper, the proposed method does not require calibration; in this way, we obtain three-dimensional camera locations without knowing any information about the parameters of the camera or even the Euclidean position of the objects in the scene. An earlier proposal was studied by Trip [9] for the case of the projective plane. Subsequently, in Refs. [10–12], the study was

extended to three-dimensional projective space using standard projective geometry. In this paper, we revisited the problem using Grassman–Cayley algebra and provide a simple method to compute the camera location. Another contribution in this work is a fast technique for automatic detection and recognition of circular coded targets inspired by previous research. Coded targets are specialized marks whose purpose is to facilitate the process of accurate point location in images used in camera calibration or similar tasks [13,14].

2. METHODOLOGY

We divide the proposed method into two main sections. In the first part, we propose a coded-target recognition technique to detect some reference points automatically for relative positioning, and the second part describes the method using Grassman–Cayley algebra to calculate the position of the camera using six reference points placed in a particular geometric configuration.

A. Detection of Reference Points

The design and recognition of artificial landmarks for reliable indoor self-localization of mobile robots is an active research area [15–17]. In this research, we use circular coded targets to identify our reference points. Note that each target has a circle in the center surrounded by a unique coded circular band. There are two main challenges to solve at the moment of identifying the coded target. First, given an image with the targets captured with a camera, the problem is how to identify single targets despite the background image. Second, we need to devise a method that identifies the ID of the coded target automatically.

We develop a technique based on [18], which is simplified here using only the ring code information by extracting the centroid of each part, and make representations with the angles formed by the centroids and the center of the coded target; see Fig. 1. Binary image processing is the first step to get rid of unnecessary information on the image; we use Otsu's method [19] to find the optimal threshold value to binarize the image. Areas of the image above the threshold become candidate targets, while all other regions are discarded as background objects if they are too large, or classified as noise if too small. Once locating the target, we use the Hough transform [20], a technique used to find instances of objects within a specific class of shapes, e.g., circles. This method also returns the coordinates of the centroid of each detected circle. When the system detects concentric circles, we consider that we have found a coded target. Then, we extract a local image window around each target. The size of the image window should be slightly larger than the ratio of the circular target, as shown in Fig. 1(a).

The images captured by the camera often present geometric distortions due to the affine transformation performed by the camera. As we do not know the orientation of the camera for the targets, the photograph will not always be taken from the front, and this produces an ellipse shape of the circular targets. After proper coordinate transformation, i.e., an inverse affine transformation, we may obtain the reconstructed coded target. Then, we remove the inner circle [Fig. 1(b)], and the target's recognition is encoded through the ring code of the target.

Fig. 1. (a) Coded target recognized after applying Otsu's binarization, (b) coded part of the target, and (c) centroids of each segment of the coded target.

As the number of targets is small, we make a database with the measures of the angles using the centroids of each white object in the final figure to the horizontal [Fig. 1(c)]. A centroid can be calculated by the arithmetic mean position of all pixels within a segmented area. For each photograph, we measure the target angles detected in the image and match the information with the database. Nevertheless, the pictures are not the same even if taken in the same way, so we give a tolerance of 5% for each angle. If there is no match with any of the points, we omit this point and go to the next coded target. Note that these coded targets are handmade and therefore are very different from each other. In general, our proposed method should work with this kind of target since most targets are simple.

After this image processing, we finally get the ID of each target so we can use the coordinates provided by the calibration board's manufacturer (measured in mm) as well as the coordinates in the plane of the picture (measured in pixels). We perform this procedure until we identify six targets and apply the following procedure to calculate the line of sight.

B. Uncalibrated Method Based on the Chasles Theorem

The method presented here has a geometric–algebraic approach. The algebraic approach is preferred to get an easier implementation within a computer system. Through Grassman–Cayley, we can represent lines in matrix form, which makes it easier to calculate it, in this case, with the Plücker coordinates. The geometric approach consists of using projective properties associated with the projection model of the camera. We make a configuration according to some projective geometry theorems in such a way as to take advantage of some properties that are invariant under projective transformations. Next, the method based on the Chasles theorem [21] and the conservation of cross-ratio in the projective transformations are given along with some considerations to take into account for this method:

- The pinhole camera model is used to model the transformation made by the camera when taking a photograph.
- We need to know at least the coordinates of six points in three-dimensional space (real world) and their corresponding projections in the photograph. We detect the coordinates of image points calculated with the image processing method described in Section 2.A (note that we are using Euclidean coordinates through the calibration grid, but this does not restrict the method since the formulation uses the projective space).

- This method uses relative positioning. In other words, we calculate the projection center according to the projective coordinate system from the reference points; in this case, the coordinate system is Euclidean.

1. Camera Model

As mention above, this method employs the pinhole camera model where each point $\mathbf{X}$ in 3D space is projected through a line that passes through the aperture in the camera. This line is known as the *line of sight* of point $\mathbf{X}$, and the camera aperture where all lines of sight intersect is known as the *center of projection* of the camera. This is the point that we want to identify, and the obtained results correspond to the coordinates of the camera in the three-dimensional world.

The camera performs a projective transformation from $\mathbb{P}^3$ to $\mathbb{P}^2$. This transformation is expressed as a linear transformation in homogeneous coordinates as shown in Eq. (1):

$$\mathbf{U} = P\mathbf{X}, \tag{1}$$

where $\mathbf{U} = [u_1, u_2, 1]^T$ is the vector of homogeneous coordinates of a point in the photo, $\mathbf{X} = [x_1, x_2, x_3, 1]^T$ is the vector of homogeneous coordinates of a point in space, and P is the matrix associated with the projective transformation.

2. Camera Projection Center

We state the problem of calculating the projection center of a photographic camera as follows.

Objective: calculate the projection center of a photographic camera.

Input: a set of six points: $\{\mathbf{X}^{[1]}, \mathbf{X}^{[2]}, \mathbf{X}^{[3]}, \mathbf{X}^{[4]}, \mathbf{X}^{[5]}$, and $\mathbf{X}^{[6]} \in \mathbb{P}^3\}$; and its corresponding projection points in a photograph: $\{\mathbf{U}^{[1]}, \mathbf{U}^{[2]}, \mathbf{U}^{[3]}, \mathbf{U}^{[4]}, \mathbf{U}^{[5]}$, and $\mathbf{U}^{[6]} \in \mathbb{P}^2\}$.

Output: coordinates of the center of projection of the camera $\mathbf{X}^{[0]} \in \mathbb{R}^3$.

We apply the following approach to calculate the position of the camera. Project all points found in the photograph towards the optical center of the photographic camera. Then, the point where all lines of sight intersect corresponds to the center of projection of the photographic camera. Hence, the approach states that to calculate the center of projection, we have to calculate the intersection of at least two lines of sight. So, we summarize the method in the following steps:

- Calculate the line of sight of point $\mathbf{X}^{[1]}$.
- Calculate the line of sight of point $\mathbf{X}^{[2]}$.
- Calculate the intersection point of both lines of sight.

The next subsection presents the method to calculate a line of sight.

3. Line of Sight of a Point

Let $\mathbf{X}^{[1]}$ be a point of $\mathbb{P}^2$, and let $\mathbf{U}^{[1]}$ be its projection in the photograph; then, we want to calculate the line of sight of point $\mathbf{X}^{[1]}$.

We need at least two known points to define any line. In this case, point $\mathbf{X}^{[1]}$ will be the first reference, and the second will be any other point lying in the same line between the center of projection of the camera and point $\mathbf{X}^{[1]}$.

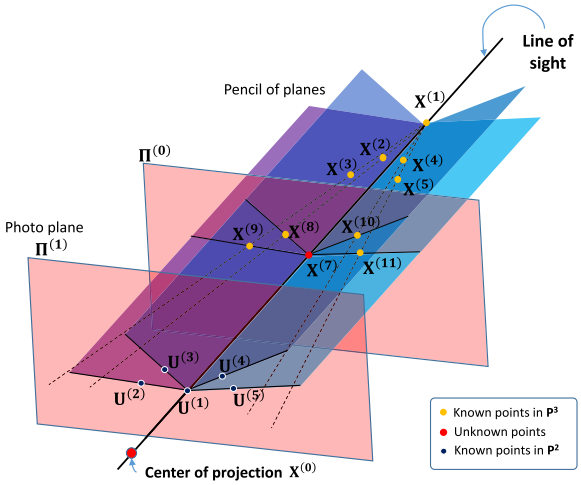

Fig. 2. Geometric configuration used to calculate the coordinates of the second point. Yellow points represent the known points given *a priori*, and red marks are those we want to know.

Something important to emphasize is that we will find the missing point using the information on the photo and a geometric configuration that takes advantage of some projective theorems. Figure 2 shows a graphic description of the proposed geometric configuration.

First we choose five of the six known points: $\mathbf{X}^{[1]}$, $\mathbf{X}^{[2]}$, $\mathbf{X}^{[3]}$, $\mathbf{X}^{[4]}$, and $\mathbf{X}^{[5]}$, and their corresponding projection points in the photograph: $\mathbf{U}^{[1]}$, $\mathbf{U}^{[2]}$, $\mathbf{U}^{[3]}$, $\mathbf{U}^{[4]}$, and $\mathbf{U}^{[5]}$. Any four of these points must be linearly independent. Hence, they must not belong to the same plane. This ensures that points $\mathbf{X}^{[1]}$, $\mathbf{X}^{[2]}$, $\mathbf{X}^{[3]}$, $\mathbf{X}^{[4]}$, and $\mathbf{X}^{[5]}$ form a base of $\mathbb{P}^3$.

We use these points to create a pencil of planes: each plane formed by the union of the line of sight and each of the other known points $\mathbf{X}^{[2]}$, $\mathbf{X}^{[3]}$, $\mathbf{X}^{[4]}$, and $\mathbf{X}^{[5]}$.

Then, we set an imaginary plane $\Pi^{[0]}$ with Eq. (2), which intersects the pencil of planes, which in turn creates a pencil of lines with the intersection of each plane with $\Pi^{[0]}$. The vertex of this pencil of lines is point $\mathbf{X}^{[7]}$, and this is the second point that will define our line of sight:

$$\Pi^{(0)} = [0, 0, 1, -\alpha]^T. \tag{2}$$

We extrapolate lines created from point $\mathbf{X}^{[1]}$ to each one of the other points until they reach plane $\Pi^{[0]}$. Using the pencil of lines created on $\Pi^{[0]}$, we can calculate the intersection with the lines corresponding to the other points, generating $\mathbf{X}^{[8]}$, $\mathbf{X}^{[9]}$, $\mathbf{X}^{[10]}$, and $\mathbf{X}^{[11]}$.

The next step to calculate the coordinates of point $\mathbf{X}^{[7]}$ consists of using some projective properties to relate plane $\Pi^{[0]}$ to photo plane $\Pi^{[1]}$. This procedure consists of calculating the cross-ratio between the points in the photograph, and a pair of conics generated in plane $\Pi^{[0]}$ with the known points $\mathbf{X}^{[8]}$, $\mathbf{X}^{[9]}$, $\mathbf{X}^{[10]}$, and $\mathbf{X}^{[11]}$ and the missing point $\mathbf{X}^{[7]}$. We explain the details of this procedure next.

According to the theorem of cross-ratio conservation, the cross-ratio is invariant under any projective transformation. Thus, the cross-ratio of the pencil of lines formed in the photograph by points $\mathbf{U}^{[1]}$ to $\mathbf{U}^{[5]}$ will be the same cross-ratio of

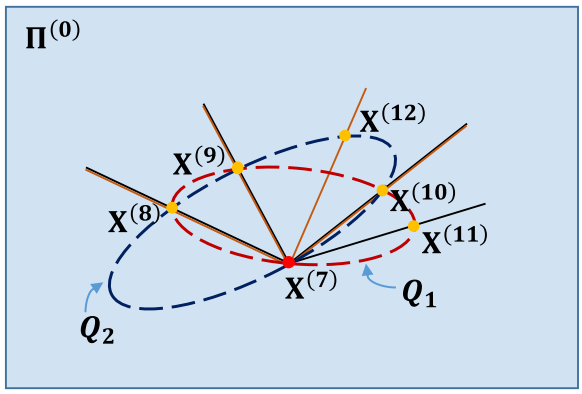

Fig. 3. Plane $\Pi^{[0]}$: conic $\mathbf{Q}_1$ is defined by the points $\mathbf{X}^{[8]}$, $\mathbf{X}^{[9]}$, $\mathbf{X}^{[10]}$, $\mathbf{X}^{[11]}$; and conic $\mathbf{Q}_2$ is defined by the points $\mathbf{X}^{[8]}$, $\mathbf{X}^{[9]}$, $\mathbf{X}^{[10]}$, $\mathbf{X}^{[12]}$. Point $\mathbf{X}^{[7]}$ is the point where conics meet.

the pencil of planes, and the same cross-ratio of the pencil of lines in $\Pi^{[0]}$. We calculate the cross-ratio of the points in the photograph using Eq. (3):

$$\lambda_1 = \frac{\left|\mathbf{U}^{(1)}\mathbf{U}^{(2)}\mathbf{U}^{(4)}\right|\left|\mathbf{U}^{(1)}\mathbf{U}^{(3)}\mathbf{U}^{(5)}\right|}{\left|\mathbf{U}^{(1)}\mathbf{U}^{(2)}\mathbf{U}^{(5)}\right|\left|\mathbf{U}^{(1)}\mathbf{U}^{(3)}\mathbf{U}^{(4)}\right|}. \tag{3}$$

According to the Chasles theorem [21], if we have a set of points that share a cross-ratio, the vertex of the pencil of lines formed by these points will lie on a conic $\mathbf{Q}_1$ that passes through all the points. By conic, we mean any curve obtained as the

Fig. 4. AICON 3D Systems GMbH calibration board.

Table 1. Point Locations Obtained from the Pictures (pixels)

	U	V
1	29.2073 ± 0.1924	243.8462 ± 0.3371
2	364.2762 ± 0.3558	48.7074 ± 0.4819
3	708.3291 ± 0.8124	241.2942 ± 0.5475
4	288.8615 ± 0.2723	28.1610 ± 0.4967
5	227.7714 ± 0.2759	242.9665 ± 0.2722
6	718.6748 ± 0.6818	460.2399 ± 0.3470

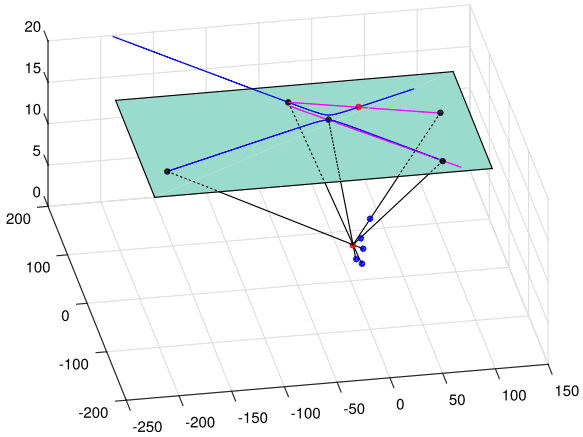

Fig. 5. Plane Π^0 where conics meet, calculated for the image in Fig. 4.

intersection of a cone with a plane, such as an ellipse, hyperbola, or parabola. Thus, all points that satisfy this relationship of cross-ratio also lies in conic $\mathbf{Q}_1$. We need to find out which one of them is the vertex of the pencil. For this, we use the remaining

Table 2. Camera Location in 3D Space (mm)

Num	X	Y	Z
1	622.2778	662.6166	1092.0335
2	623.5819	662.2746	1090.2416
3	623.0528	662.4133	1090.9680
4	616.6402	652.6795	1029.7691
5	640.3290	666.0873	1117.5522
6	631.6202	673.7676	1050.4442
7	615.1859	681.7467	1023.2794
8	653.7578	663.2008	1084.2768
9	640.8150	666.4067	1119.6247
10	649.8965	648.5087	1140.2568
11	637.3536	627.2716	1214.0029
12	657.3513	664.2196	1089.5548
13	640.6168	666.2764	1118.7794
14	642.8413	689.8115	994.8887
15	669.3052	638.5945	1174.1453
16	655.8245	663.7867	1087.3126
17	648.18012	671.2482	1151.0446
18	622.5441	660.7908	1095.3794
19	625.2033	661.1222	1097.1414
20	623.6545	660.9347	1096.1445
21	641.7108	665.4089	1119.9518
22	626.1723	665.9782	1077.4168
23	620.3564	669.0393	1067.7694
24	620.9494	653.9010	1036.0973
25	640.4023	665.7748	1121.9022
26	631.6046	657.8524	1108.3184
27	627.4130	651.6987	1128.4809
28	622.0105	654.2018	1037.6556
29	640.9202	665.6299	1121.1302
30	630.0178	671.4766	1058.3774
Mean	634.7196	662.1573	1094.4647
Std Dev	13.6882	11.6284	46.0239

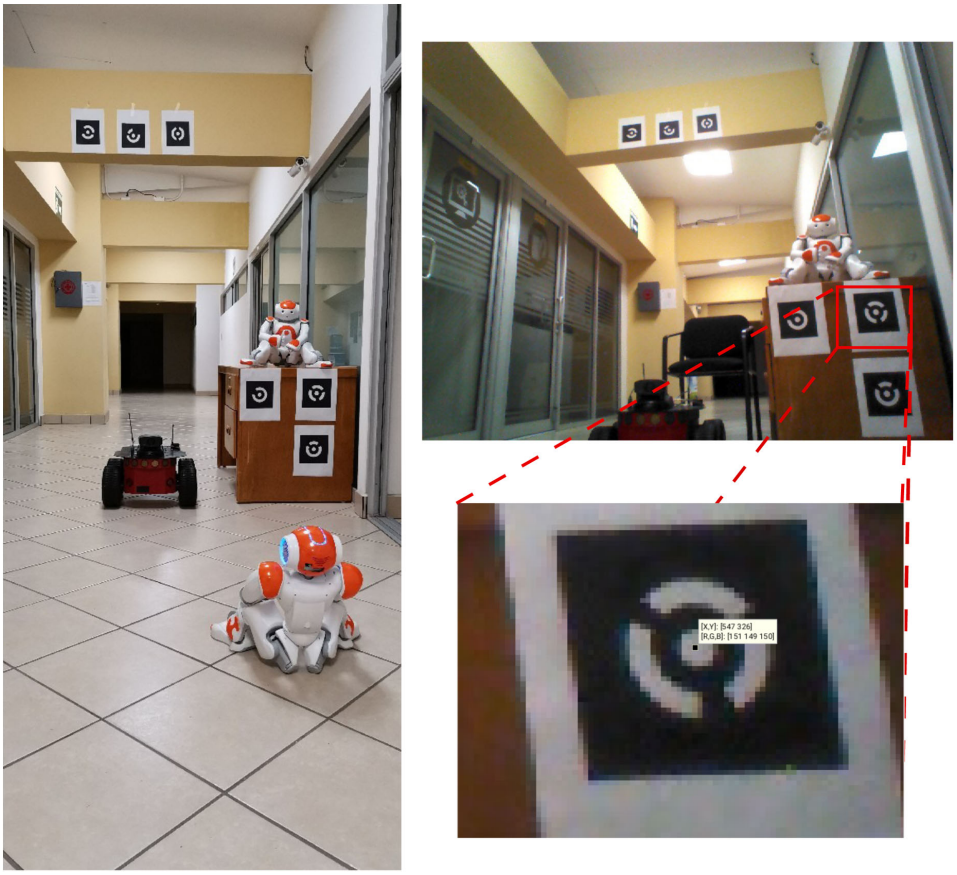

Fig. 6. Self-localization proposal for indoor robot localization using artificial visual landmarks.

known point $\mathbf{X}^{[6]}$ and ignore any of the other points, to form new sets of pencil-of-planes, pencil-of-lines, and a new conic, as shown in Fig. 3.

Next, we calculate the intersection between conics $\mathbf{Q_1}$ and $\mathbf{Q_2}$ whose coordinates define point $\mathbf{X}^{[7]}$. The remaining steps consist of calculating another line of sight, and the intersection of these lines is the projection center of the camera; in other words, the position of the camera (in our case, Euclidean) in the chosen coordinate system.

3. EXPERIMENTS

In this experiment, we use an AICON 3D system GMbH calibration board, a flat panel of carbon fibers with reflective targets with known coordinates provided by the manufacturer; the grid is shown in Fig. 4. Some of the targets are coded, so we can recognize a specific point. For this problem, we use only coded targets as reference points.

Targets are in two different planes that have a separation of 200 mm. Some of them are at the level of the panel, and five of them are suspended on poles over the flat panel, ensuring that we are using points in different planes. With the method described in Section A, we successfully identify six reference points and therefore the coordinate given in pixels in the picture; see Table 1. We take photographs with a Pulnix TM-9701d camera with a Fujinon, HF16A-2M1, of focal length $f = 16$ mm.

Table 3. Point Locations Obtained with the NAO Robot (pixels)

	U	V
1	560.7333 ± 0.3407	424.5333 ± 0.3924
2	547.5333 ± 0.2249	326.2667 ± 0.2537
3	$452.8500 \pm 0.0.2980$	335.0333 ± 0.1269
4	343.9833 ± 0.0913	98.5500 ± 0.2013
5	298.2667 ± 0.2537	106.5500 ± 0.2403
6	252.8500 ± 0.3693	110.4900 ± 0.1647

We observe in Fig. 5 the lines of sight for each point in the image, as well as the conics formed on plane $\Pi^{[0]}$. As we can appreciate, the conics formed in this plane correspond to a parabola and a hyperbola. We repeat the experiment with 30 images of the calibration grid shown in Fig. 4. Table 2 shows the average errors and standard deviation on each axis of the calculated optical center. As we can observe from the results, the proposed method for coded target recognition works, and the accuracy is good enough considering robotics tasks.

4. TESTING THE PROPOSAL IN THE REAL WORLD

In this section, we address the localization problem for a humanoid robot in an indoor environment with the proposed method.

Table 4. Comparison Between Calibrated and Uncalibrated Methods

	Faugeras–Toscani			Proposed Method		
	X	Y	Z	X	Y	Z
1	64.65	61.42	−166.38	76.87	52.86	−170.68
2	58.65	53.12	−147.95	81.75	59.76	−180.75
3	91.33	68.61	−277.31	87.72	37.98	−192.83
4	77.34	63.75	−219.29	92.55	54.59	−202.84
5	61.68	55.55	−160.74	81.99	51.70	−181.26
6	48.82	50.64	−111.12	76.02	43.04	−169.52
7	68.62	62.52	−193.40	84.97	47.69	−188.74
8	71.60	60.53	−198.60	83.32	51.82	−183.98
9	58.22	53.86	−145.49	81.54	69.05	−180.13
10	61.85	54.70	−166.78	82.12	64.74	−181.83
11	58.65	53.12	−147.95	81.75	42.81	−180.38
12	72.00	60.35	−199.49	83.27	66.15	−183.73
13	49.41	48.97	−117.96	77.30	53.86	−172.24
14	58.65	53.12	−147.95	81.75	49.66	−180.38
15	69.84	61.90	−193.02	82.78	53.80	−183.48
16	50.14	47.97	−116.28	76.74	48.91	−169.98
17	58.34	55.64	−145.06	81.29	49.34	−180.01
18	50.23	49.13	−118.12	77.51	57.93	−172.22
19	65.41	58.75	−177.35	82.31	57.50	−182.46
20	54.21	52.32	−124.89	80.48	57.54	−177.69
21	62.28	54.76	−166.74	82.23	53.57	−181.84
22	79.33	66.41	−224.55	86.28	43.57	−190.09
23	67.36	59.69	−175.86	82.63	53.82	−182.24
24	66.02	57.91	−180.81	82.83	58.68	−182.72
25	51.21	48.61	−120.67	77.30	52.60	−171.28
26	67.37	57.82	−180.83	82.72	55.51	−182.44
27	70.85	61.97	−192.04	83.02	53.87	−183.45
28	67.06	59.44	−182.53	85.20	48.38	−188.23
29	63.13	54.82	−165.77	82.47	51.56	−181.81
30	59.95	52.30	−148.71	81.48	45.81	−180.66
Mean	63.47	56.66	−167.12	82.01	52.94	−181.33
Std Dev	9.61	5.39	37.09	3.45	6.92	7.12

NAO (pronounced now) is the name of a humanoid robot dedicated to research, education, and entertainment purposes that we use in the experiments. Figure 6 illustrates the setup where multiple markers placed on the hall at the EvoVisión laboratory feed the calibrated and uncalibrated approaches. The application of artificial landmarks for camera robot localization has several advantages, mainly for the case of known environments. It is easy to design landmark recognition algorithms with information about the position of each coded target. Also, such a strategy avoids problems of historical localization data, and it does not have the accumulative error of markerless systems. This section includes a comparison with the calibrated method of Faugeras and Toscani [5]. As the first step, we compute the localization of six targets in the known environment; see Fig. 6. Table 3 provides the average image coordinates obtained from 30 photographs taken with the NAO robot. Then, we calculate the optical center with the uncalibrated and calibrated approaches; see Table 4. The results confirm that the system can provide stable and accurate position estimation measurements for the NAO humanoid robot.

5. CONCLUSION

In this paper, we presented a method to calculate the position of a camera. One of its most important features is that it does not require calibration of the camera. Therefore, it is possible to zoom in, change the focus, move the camera, or any other actions involved in the active vision process. The algorithm described in this paper follows a simple strategy, and we successfully apply it to real images. Results obtained give us the reliability to implement this method in a mobile robot, guaranteeing that the integrity of the robot is uncompromised by errors in the calculation of its position.

Funding. Centro de Investigación Científica y de Educación Superior de Ensenada, Baja California (634-135).

Acknowledgment. This research was partially funded by CICESE through the project 634-135, "Estudio de la programación cerebral en problemas de reconocimiento a gran escala y sus aplicaciones en el mundo real." The first author graciously acknowledges the scholarship paid by the Consejo Nacional de Ciencia y Tecnologa (CONACyT).

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. L. Payá, A. Gil, and O. Reinoso, "A state-of-the-art review on mapping and localization of mobile robots using omnidirectional vision sensors," J. Sens. **2017**, 1–20 (2017).
2. J. Wang and Y. Takahashi, "Indoor mobile robot self-localization based on a low-cost light system with a novel emitter arrangement," ROBOMECH J. **5**, 17 (2018).
3. S. Davis, K. G. Ricks, and R. A. Taylor, "Reflective fiducials for localization with 3D light detection and ranging scanners," IEEE Access **7**, 45291–45300 (2019).
4. Y. Wu, F. Tang, and H. Li, "Image-based camera localization: an overview," Vis. Comput. Ind. Biomed. Art **1**, 8 (2018).
5. O. Faugeras and G. Toscani, "Camera calibration for 3D computer vision," in *International Workshop on Machine Vision and Machine Intelligence*, Tokyo, Japan, 1987, pp. 240–247.
6. O. Faugeras, Q.-T. Luong, and T. Papadopoulou, *The Geometry of Multiple Images: The Laws That Govern the Formation of Images of a Scene and Some of Their Applications*, s(MIT, 2001).
7. F. Liu, J. Zhang, J. Wang, and B. Li, "The indoor localization of a mobile platform based on monocular vision and coding images," Int. J. Geo-Inf. **9**, 21 (2020).
8. W. Li, S. Shan, and H. Liu, "High-precision method of binocular camera calibration with a distortion model," Appl. Opt. **56**, 2368–2377 (2017).
9. C. Tripp, "Where is the camera? The use of a theorem in projective geometry to find from a photograph the location of the camera," Math. Gaz. **71**, 8–14 (1987).
10. L. Morin, "Quelques contributions des invariants projectifs à la vision par ordinateur," Ph.D. thesis (Institut National Polytechnique de Grenoble, 1993).
11. R. Mohr and L. Morin, "Relative positioning from geometric invariants," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 1991), pp. 139–144.
12. R. Mohr and L. Morin, "Geometric solutions to some 3D vision problems," in *Geometric Reasoning for Perception and Action* (1993, pp. 139–162.
13. W. Li, G. Liu, L. Zhu, X. Li, Y. Zhang, and S. Shan, "Efficient detection and recognition algorithm of reference points in photogrammetry," Proc. SPIE **9896**, 1–10 (2016).

14. S. J. Ahn and W. Rauh, "Circular coded target for automation of optical 3D-measurement and camera calibration," Int. J. Pattern Recognit. Artif. Intell. **15**, 905–919 (2001).
15. X. Zhong, Y. Zhou, and H. Liu, "Design and recognition of artificial landmarks for reliable indoor self-localization of mobile robots," Int. J. Adv. Robot. Syst. **14**, 1729881417693489 (2017).
16. A. R. Cantieri, R. F. Rohrich, A. S. Oliveira, J. A. Fabro, and M. A. Wehrmeister, "Collaborative quadricopter-mobile robots ground scan using ARTAGS visual pose estimation," in *Latin American Robotics Symposium (LARS) and Brazilian Symposium on Robotics (SBR)* (2017), pp. 1–6.
17. Y. Li, S. Zhu, Y. Yu, and Z. Wang, "An improved graph-based visual localization system for indoor mobile robot using newly designed markers," Int. J. Adv. Robot. Syst. **15**, 1729881418769191 (2018).
18. C. Guo, X. Cheng, H. Cui, N. Dai, and J. Weng, "A new technique of recognition for coded targets in optical 3D measurement," Proc. SPIE **9276**, 92761H (2014).
19. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern. A **9**, 62–66 (1975).
20. P. Hough, "Method and means for recognizing complex patterns," US Patent 3,069,654 (18 September 1962).
21. M. Chasles, *Traité des sections coniques: faisant suite au Traité de géométrie supérieure* (Gauthier-Villars, 1865), Vol. **1**.