

# Fractal Markers: A New Approach for Long-Range Marker Pose Estimation Under Occlusion

FRANCISCO J. ROMERO-RAMIRE<sup>1</sup>, RAFAEL MUÑOZ-SALINAS<sup>1,2</sup>,  
AND R. MEDINA-CARNICER<sup>1,2</sup>

<sup>1</sup>Departamento de Informática y Análisis Numérico, Edificio Einstein, Campus de Rabanales, Universidad de Córdoba, 14071 Córdoba, Spain

<sup>2</sup>Instituto Maimónides de Investigación Biomédica de Córdoba (IMIBIC), 14004 Córdoba, Spain

Corresponding author: Rafael Muñoz-Salinas (rmsalinas@uco.es)

This work was supported by the Spain Ministry of Economy, Industry and Competitiveness, and FEDER under Project TIN2016-75279-P and Project IFI16/00033 (ISCIII).

**ABSTRACT** Squared fiducial markers are a powerful tool for camera pose estimation in applications such as robots, unmanned vehicles and augmented reality. The four corners of a single marker are enough to estimate the pose of a calibrated camera. However, they have some limitations. First, the methods proposed for detection are ineffective under occlusion. A small occlusion in any part of the marker makes it undetectable. Second, the range at which they can be detected is limited by their size. Very big markers can be detected from a far distance, but as the camera approaches them, they are not fully visible, and thus they can not be detected. Small markers, however, can not be detected from large distances. This paper proposes solutions to the above-mentioned problems. We propose the Fractal Marker, a novel type of marker that is built as an aggregation of squared markers, one into another, in a recursive manner. Also, we proposed a novel method for detecting Fractal Markers under severe occlusions. The results of our experiments show that the proposed method achieves a wider detection range than traditional markers and great robustness to occlusion.

**INDEX TERMS** Fiducial markers, marker mapping, pose estimation.

## I. INTRODUCTION

Camera pose estimation is a common problem in many applications. Solutions using natural features have attracted most of the research effort, reaching a high degree of performance [1], [2]. Nevertheless, they have several limitations in some realistic scenarios. First, when using a single camera, the obtained pose is not on the real scale. Second, they require a certain amount of texture, which in some indoor environments is not available (e.g., labs and corridors). Third, their detection and identification can be very time-consuming.

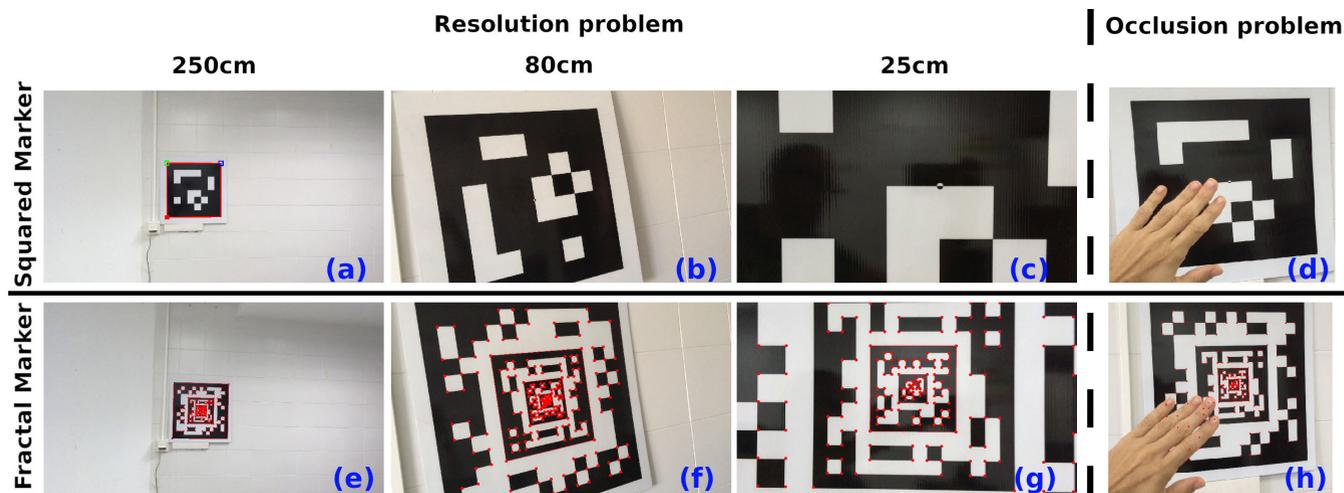
In some use cases, it is possible to place artificial landmarks to ease the pose estimation task and to solve the above-mentioned problems. In particular, squared fiducial markers have become very popular for that purpose [3]–[7]. They are composed by an external black border, that can be easily detected in the environment, and a inner binary pattern that uniquely identify them (see Fig 1d). Their main advantages are three. First the camera pose can be obtained in the correct scale by using only its four external corners. Second, their

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaogang Jin.

detection is extremely fast using low CPU usage [8]. Finally, their detection is robust to light and perspective transforms.

For these reasons, their use has spread in a wide variety of fields, such as surgery [9]–[11], distributed autonomous 3D printing [12], human-robot interaction [13], autonomous aerial vehicle landing [14], [15], patient positioning in radiotherapy treatments [16], study animal behaviour [17], human cognitive processes [18], 3D body scanning [19], [20], robotic grasping [21], underwater manipulation [22], etc.

Despite the many advantages of fiducial markers, their use in pose estimation has three main drawbacks. First, due to the fixed size of the marker, there is an intrinsic limitation in the range of possible distances at which it can be detected. We call this the *resolution problem* and is shown in Fig. 1(a-c). The second problem is the *occlusion problem*. Most marker detection methods are incapable of dealing with occlusions and those that deal with it are very slow (see Fig. 1d). Third, estimating the camera pose using only the four most external corners discard important information about the inner marker structure that can be exploited to improve the precision of the pose [23]. This is the rationale behind another kind of planar structured markers, such as the



**FIGURE 1.** Common problems of squared markers: the resolution problem (a-c) and the occlusion problem (d). Fig. (a-c) show a squared marker observed at the distances 250 cm, 80 cm and 25 cm from the camera and overlaid as red rectangles the ArUco [4] detections (only works in the first case). Under the same conditions, Fig. (e-g) show the results of our proposal, the Fractal Marker, overlaying in red color the inner marker corners detected. Fig. (d,h) show the results in case of occlusion of both methods. As can be seen, Fractal Markers can be detected in more cases than regular squared markers.

chessboards patterns commonly used for calibration tasks in popular tools such as OpenCV [24].

This paper proposes a novel type of marker, the *Fractal Marker* (Fig. 1f), designed as the composition of squared fiducial markers of different sizes, one into another. As shown in Fig. 1(e-g), the proposed Fractal Marker can be detected from a wider range of distances than a single marker. Also, it alleviates the partial occlusion problem, since the pose can be estimated from any marker even if the most external one is occluded (Fig. 1(g,h)). Nevertheless, in order to be fully robust against occlusion, the second contribution of this paper is a novel method for marker tracking able to find the marker (and estimate the pose) by detecting and classifying its inner corners. Therefore, our method is not only capable of detecting the marker in case of occlusion, but it is also able to estimate the pose more precisely by taking advantage of all the corner information available into the marker.

As our experiments show, our approach achieves a wider detection range than traditional markers and high robustness to occlusion, while adding little computational cost. The proposed method is a step forward for the use of fiducial markers that allow expanding their use to applications where only a partial view of the marker is expected, or it must be detected from a wide range of distances, such as augmented reality applications where interaction causes frequent occlusion of the marker, or drone landing tasks where the marker must be detected at a very large range of distances.

The remainder of this work is organized as follow. Section II reviews the related works, while Section III explains the design of Fractal Markers and Section IV describes the proposed method for pose estimation using them. Finally, Section V shows the experimentation carried out and Section VI draws some conclusions.

## II. RELATED WORKS

As previously indicated, fiducial markers are a very popular method for pose estimation, and several approaches have been proposed. ARToolKit [25] is one of the first square-based fiducial markers systems. It is composed by a set of valid image patterns inside a wide black square. Despite its success, it presents several limitations. Their matching method presents both high false positive rates and inter-marker confusion rates. ARToolKit Plus [26] tries to solve its deficiencies by employing a binary BCH code [27] that provides a robust detection and correction. Nevertheless, the project was finally halted and followed by Studierstube project [28].

BinARyID [29] uses a method to generate customizable binary-coded markers instead of using a pre-defined dataset. However, the system does not consider possible errors in the detection and correction. Nevertheless, these aspects are considered by AprilTags [5] which introduces methods for correction.

ArUco [4] proposes a robust method for markers detection. It uses an adaptive thresholding method which is robust to different illumination conditions and performs error detection and correction of the binary codes implemented. Also, ArUco presents a method to generate markers that maximizes the inter-marker distance and the number of bit transitions, using Mixed Integer Linear Programming [30].

A recent work [8] introduces improvements allowing to speed up the computing time in video sequences by wisely exploiting temporal information and an applying a multiscale approach.

Despite the significant advances achieved so far, fiducial markers have some limitations. First, if the marker is partially occluded, pose estimation cannot be done. Second, the fixed

size of the marker makes it impossible to detect them under a wide range of distances.

Some authors have proposed alternatives to overcome the above problems. The ArUco library partially solves the occlusion problem by using multiple markers creating what they call *board*. A board is a pattern composed of multiple markers and all of them referred to the same reference system.

On the other hand, ARTag [3] handles the partial occlusion using an edge based method. Edge pixels are thresholded and connected in segments, which are grouped into sets and used to create a mapping homography. Nevertheless, markers can not be detected when more than one edge is occluded and their is very slow.

Another approach to alleviate the occlusion problem is proposed by Alvarez et al. [31]. The authors propose a type of markers with textured and coloured borders. The system has a database of descriptors of the patterns, which are used in case of occlusion. Their approach have several limitations though. First, marker generation is a complex process requiring an offline process to create a database of SIFT keypoint descriptors. Second, they do not deal with the problem of detecting the marker under a wide range of distances.

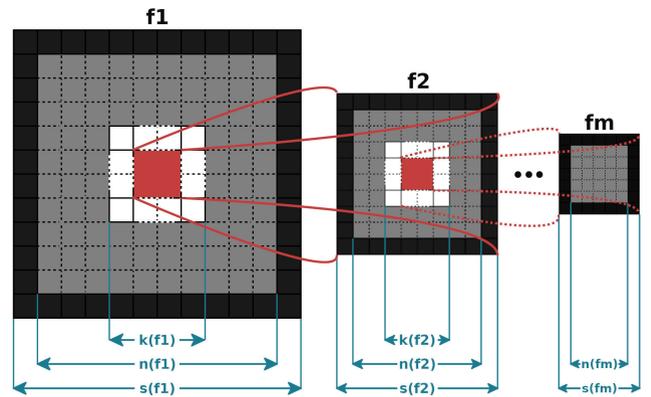
Another very popular library is Apriltag3 [32], which introduces a new configurable marker concept that allows employing recursive patterns. Although in theory their system could be adapted to solve the same problems we are solving in this paper, they do not show deal with them in their publication.

Finally, HArCo [33] is the work most the related to ours. The authors propose a new hierarchical marker structure. Assuming that small pixel changes in the cells of a traditional marker do not change the detection and identification of markers, white cells are replaced by new layers of sub-markers. HArCo system uses the same methodology proposed by ArUco for the individualized detection of the markers that compose the hierarchical marker, and the final pose estimation is given by the mean of the positions provided by all the markers correctly detected. Unfortunately the HArCo system is not available for public use and consequently it is not possible to compare against it.

This work proposes the *Fractal Marker* as an alternative to overcome the occlusion and resolution problems. Multiple markers are used sharing the same reference point. Unlike the marker board where the markers are displaced at different distances from the common center, our method proposes that there is no displacement. For this it is necessary to use markers of different sizes that can be configured, giving the appearance of a recursive marker.

### III. FRACTAL MARKER DESIGN

Let us define a Fractal Marker  $F$  as a set of  $m$  squared markers  $(f^1, f^2, \dots, f^m)$ , placed one into the another in a recursive manner (see Fig. 2). In a Fractal Marker, each squared marker  $f^i$  is comprised by an external black border (for fast detection), a region reserved for bit identification (shown in grey), and a white region surrounding its inner marker  $f^{i+1}$ . This white band is necessary to ease the detection of the



**FIGURE 2.** Generic structure of Fractal Marker  $F$ , in which each marker is composed of a set of cells that can be grouped into three categories. The black band correspond to the marker border, the gray cells configure and uniquely determine the marker, and finally, the white band facilitate the detection of the inner marker.

inner marker black border. This section explains the proposed design to generate Fractal Markers.

Let denote  $s(f^i)$ ,  $n(f^i)$  and  $k(f^i)$  the length side of the black region, the identification region (shown in gray) and the white region, respectively, shown Fig. 2, for a squared marker  $f^i$ . There is an exception for the most internal marker  $f^m$ . In this case, the white region will not be necessary because no marker will be placed inside it, i.e.,  $k(f^m) = 0$ . Notice that these values are calculated with regard to the reference system with origin in the bottom left external corner of the internal marker  $f^i$ .

Formally speaking, the only restrictions for the values of  $s(f^i)$ ,  $n(f^i)$  and  $k(f^i)$  are:

$$s(f^{i+1}) < k(f^i) \forall i \neq m,$$

and

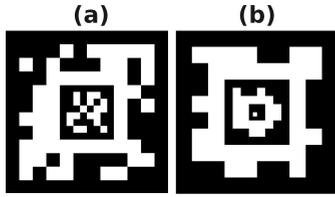
$$k(f^i) < n(f^i) < s(f^i) \forall i.$$

Each marker  $f^i$  can have a different number of bits for region identification depending on the area of its identification region (of length  $n(f^i)$ ). Please notice that the number of bits in the identification region of  $f^i$  is less than in a traditional squared fiducial marker.

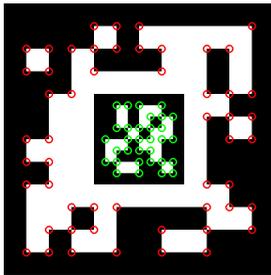
Then, the size of region codification of internal markers  $f^i$ ,  $i \in \{1, \dots, m\}$  is (see Fig. 2):

$$S_R(f^i) = n(f^i)^2 - k(f^i)^2. \tag{1}$$

Fig. 3 shows two different possible combinations of internal markers for a Fractal Marker. Fig. 3a shows a Fractal Marker composed of two internal markers  $s(f^1) = 12$ ,  $n(f^1) = 10$ ,  $k(f^1) = 6$ ,  $S_R(f^1) = 64$  and  $s(f^2) = 8$ ,  $n(f^2) = 6$ ,  $k(f^2) = 0$ ,  $S_R(f^2) = 36$ . In Fig. 3b, the Fractal Marker is composed of three internal markers  $s(f^1) = 10$ ,  $n(f^1) = 8$ ,  $k(f^1) = 6$ ,  $S_R(f^1) = 28$ ;  $s(f^2) = 8$ ,  $n(f^2) = 6$ ,  $k(f^2) = 4$ ,  $S_R(f^2) = 20$  and  $s(f^3) = 4$ ,  $n(f^3) = 2$ ,  $k(f^3) = 0$ ,  $S_R(f^3) = 4$ .



**FIGURE 3.** Examples of different configurations of Fractal Marker and areas of identification region  $S_R(f^i)$ . (a) Fractal Marker composed of two internal markers  $F = \{f^1, f^2\}$ , whose identification areas are  $S_R(f^1) = 64$  and  $S_R(f^2) = 36$ . (b) Fractal Marker composed of three internal markers  $F = \{f^1, f^2, f^3\}$ , whose identification areas are  $S_R(f^1) = 28$ ,  $S_R(f^2) = 20$ ,  $S_R(f^3) = 4$ .



**FIGURE 4.** Fractal Marker composed of two internal markers. The inner corners of marker  $f^1$  and  $f^2$  are shown in red and in green respectively.

The selected configuration depends on the needs of the application. The more internal markers are employed, the larger the operating range of the Fractal Marker.

Let us denote

$$\text{bits}(f^i) = (b_1^i, \dots, b_j^i, \dots, b_{S_R(f^i)}^i), \quad (2)$$

where  $b_j^i \in \{0, 1\}$ ,  $\forall j = 1, \dots, S_R(f^i)$ , to the information bits of marker  $f^i$ . Notice that the bit sequence is created row by row starting from the top-left bit (see Fig.5). The inner bits of a Fractal Marker are randomly generated using a Bernoulli distribution (i.e.,  $b_j^i \sim \text{Be}(1/2)$ ). However, not any configuration randomly obtained can be considered valid because some of them are identical under rotation. To avoid that, a randomly generated marker is considered valid when the Hamming distance in its three possible rotations is greater

than zero, i.e.:

$$H(\text{bits}(f^i), \text{bits}(R_j(f^i))) > 0, \quad \forall j \in \{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\}, \quad (3)$$

where  $H$  is the Hamming distance between two markers, and  $R_j$  is a function that rotates the marker matrix  $f^i$  in the clockwise direction a total of  $j$  degrees (see Fig. 5). If Eq 3 is not fulfilled, then the marker  $f^i$  is not valid and the process of randomly selecting bits is repeated until a valid marker  $f^i$  is obtained. A Fractal Marker  $F$  is valid when all inner markers  $f^i$  are valid.

Marker detection and pose estimation is based on detecting and analyzing the projection the marker corners in the image. Let us denote the three-dimensional coordinates of the *four external corners* of  $f^i$  as w.r.t. the marker center as:

$$\begin{aligned} c_1^i &= (s(f^i)/2, -s(f^i)/2, 0) \\ c_2^i &= (s(f^i)/2, s(f^i)/2, 0) \\ c_3^i &= (-s(f^i)/2, s(f^i)/2, 0) \\ c_4^i &= (-s(f^i)/2, -s(f^i)/2, 0) \end{aligned} \quad (4)$$

We are assuming that the marker is printed on a planar surface, thus, the third component is zero for all the corners.

In addition to four external corners  $c_j^i \in \mathbb{R}^3$  (Eq. 4) of each marker  $f^i$ , there is a set of internal corners (see Fig. 4) that can be wisely employed for marker tracking in case of occlusion, and also refine the pose.

Let us denote as  $W^i$  the set of *internal corners* of marker  $f^i \in F$ :

$$W^i = (w_1^i, \dots, w_n^i), w_j^i \in \mathbb{R}^3$$

where  $w_j^i$  are the three-dimensional coordinates as w.r.t. the marker center. Fig. 4 shows an example of a Fractal Marker composed by two markers  $f^1$  and  $f^2$  where their internal corners have been depicted as red and green coloured circles, respectively. Please notice that *four external corners* of markers are not included as *internal corners* for any marker.

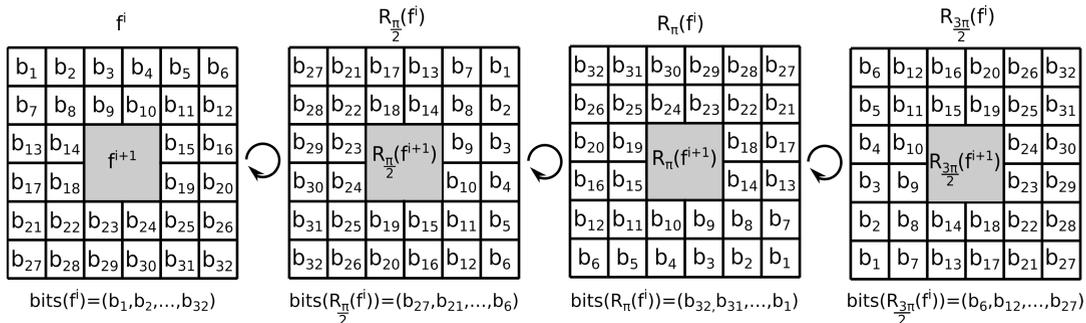
Finally, let us denote

$$C^i = \{\{W^i, c_1^i, c_2^i, c_3^i, c_4^i\},$$

to the set of internal and most external corners of each marker  $f^i \in F$ , and

$$C(F) = \{\{C^i\}/f^i \in F\}$$

to the set of all the marker corners of a Fractal Marker  $F$ .



**FIGURE 5.** Four possible rotations of a marker  $f^i$ .

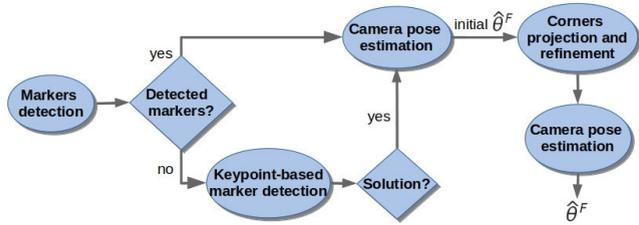


FIGURE 6. General workflow of proposed method for marker pose estimation.

#### IV. FRACTAL MARKER DETECTION

This section explains the proposed method for detecting and tracking Fractal Markers under occlusion. Fig. 6 depicts the workflow of our method. The first step of the process is to detect markers (Section IV-A). If at least one marker is detected, the detected corners are used to obtain an initial estimation of the marker pose (Section IV-B), which is employed to project the expected location of the Fractal Marker corners  $\mathcal{C}(F)$  in the image. The projected locations are used as the starting point for a refinement process to accurately find their location in the image. The whole set of refined corners are then used to compute again the marker pose, which now contains more points and thus obtains a more precise location (Section IV-C).

If no markers are detected in the initial step, our method aims at detecting the marker location using the previous detection as the starting point. To do so, the FAST [34] corner detector is employed to extract all the relevant corners in the image. The corners are then classified into the three categories (explained in Sect. IV-D). Then, a novel method for matching the observed corners with the marker corners  $\mathcal{C}(F)$  using the RANSAC algorithm is employed. As a result, our method is able to obtain an initial marker pose. At this point, this branch of the workflow merges to the other one in the “corner projection” step, in order to obtain a refined marker pose (Section IV-D).

This section provides a detailed explanation of the different steps involved in the process.

##### A. MARKERS DETECTION

The first step of the process is trying to detect the markers  $f^i$  that compose the Fractal Marker. This process is the same employed in [4] and is only able to extract the most external corners  $c_j^i$  of a marker  $f^i$ . To do so, the following steps are employed :

###### 1) IMAGE SEGMENTATION

A Fractal Marker is composed of several squared-based markers which have a black border surrounded by a white space that facilitates its detection. The method uses a local adaptive threshold which makes a robust detection regardless of light conditions (Fig. 7b).

###### 2) CONTOUR EXTRACTION AND FILTERING

Contour extraction of each internal marker is performed by Suzuki and Abe [35] algorithm. It provides a set of contours,

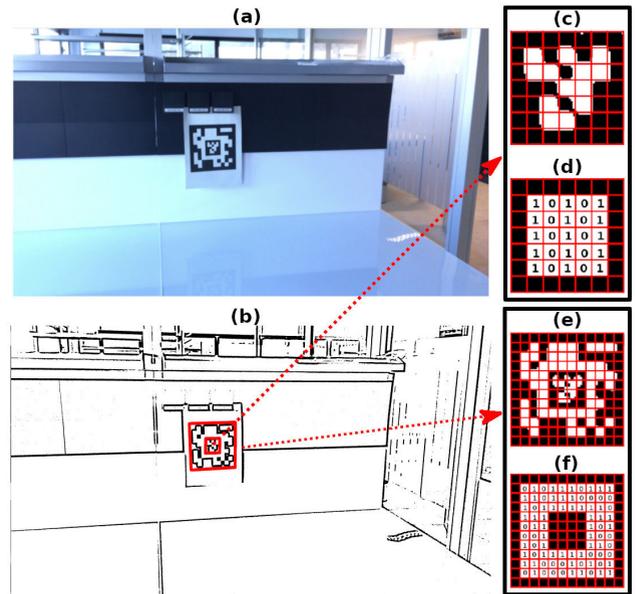


FIGURE 7. Detection and identification of Fractal Markers. (a) Original image. (b) Thresholded image showing the result of contour extraction and filtering. (c and e) Canonical images of rectangular contours containing our markers. (d and f) Binarized versions of the canonical images.

many of which correspond to unwanted objects. A filtering process is carried out using Douglas and Peucker algorithm [36] which selects only the ones most similar to a polygon (Fig. 7b).

##### 3) MARKER CODE EXTRACTION

The next step consists in analyzing the inner region of the remaining contours to determine which of them are valid markers. First, it is necessary to remove perspective projection (using a homography transform) and subsequently thresholded using Otsu’s method [37]. The resulting image is divided into a regular grid and each element is assigned the value 0 or 1 depending on the values of the majority of pixels (Fig. 7(c-f)) Finally, it is necessary to compare the candidate marker with a set of valid markers. Four possible comparisons of each candidate are made, corresponding to the four possible orientations.

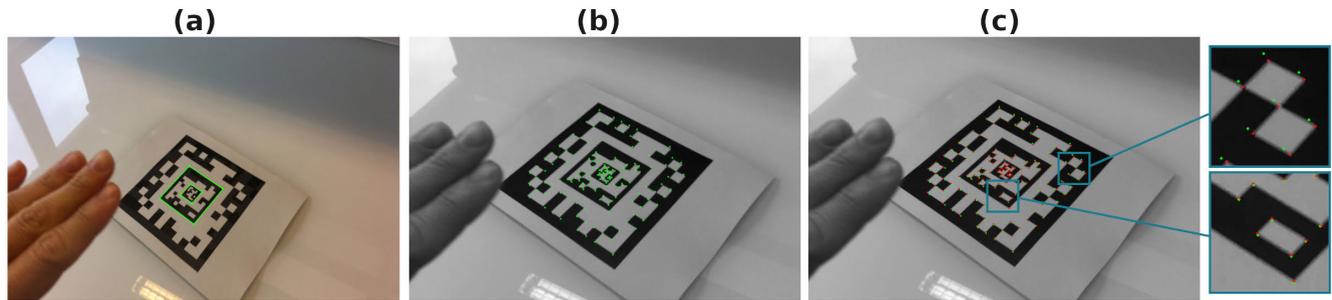
As a result of the process, an initial set of external marker corners  $\mathcal{C}'$  belonging to the external black borders is obtained. An initial pose can be obtained from them as explained later in Section IV-B.

##### B. MARKER POSE ESTIMATION

Let us define the pose of a marker  $\theta \in \mathbb{R}^6$  by its three rotational and translational components  $r = (r_x, r_y, r_z)$  and  $t = (t_x, t_y, t_z)$ :

$$\theta = (r, t) \mid r, t \in \mathbb{R}^3 \quad (5)$$

Using Rodrigues’ rotation formula, the rotation matrix  $\mathbf{R}$  can be obtained from  $r$ .



**FIGURE 8.** (a) Detection of markers and external corners in original image. (b) Initial estimation of the position using external corners of the detected markers. (c) Refinement of the pose estimation: the green points represent the estimate of the previous step (b), in red the new estimation.

A point  $p \in \mathbb{R}^3$  projects into the camera plane into a pixel  $u \in \mathbb{R}^2$ . Assuming that the camera parameters are known, the projection can be obtained as a function:

$$u = \Psi(\delta, \theta, p), \tag{6}$$

where

$$\delta = (f_x, f_y, c_x, c_y, k_1, \dots, k_n),$$

refers to the camera intrinsic parameters, comprised by the focal distances  $(f_x, f_y)$ , optical center  $(c_x, c_y)$  and distortion parameters  $(k_1, \dots, k_n)$  [24].

Then, marker pose estimation is the problem of minimizing the reprojection error of the observed marker corners:

$$\hat{\theta} = \arg \min_{\theta} \sum_{p \in \mathcal{D}} [\Psi(\delta, \theta, p) - O(p)]^2 \tag{7}$$

where  $O(p) \in \mathbb{R}^2$  is the observed position in the camera image of corner  $p \in \mathcal{D}$ . The corner set  $\mathcal{D}$  can have any type of corners (i.e., external and internal corners).

When all the points lay in the same plane, it is a special case that can be solved using specific methods such as the Infinitesimal Plane-Based Pose Estimation (IPPE) [38].

### C. CORNER PROJECTION AND REFINEMENT

Once an initial estimation of the marker pose is obtained from a reduced set of corners  $\mathcal{C}'$ , it is possible to find all the visible corners and use them to refine the pose even further. To do so, first, all the marker in  $\mathcal{C}(F)$  are projected (Eq. 6) on the camera image. Then their location is refined up to subpixel accuracy. Finally, the refined corner locations are employed then to obtain a refined pose using again Eq. 7.

Subpixel corner refinement consists in analyzing a small squared region of length  $s_{min}$  around the corner location to find the maxima of the derivative within the region. In smaller images, the region of analysis becomes smaller and thus the computing time is greatly reduced. Consequently, the corner refinement process is done as a multiscale process using an image pyramid of the original image. We start by finding, for each corner, the smaller image of the pyramid at which the corner can be first refined. After an initial refinement, its

location is refined again in the next (and larger) image of the pyramid. The process is repeated until the corner is finally refined in the original image.

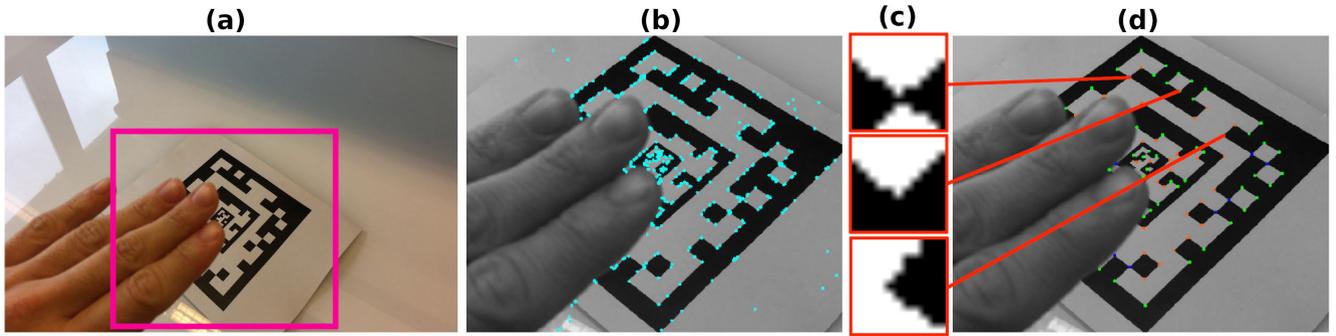
Let us denote  $\mathcal{I} = (I^0, I^2, \dots, I^P)$  as the image pyramid, where  $I^0$  is the original image, which is scaled using a scale factor of two. For each marker, we select the initial image in the pyramid  $I^j \in \mathcal{I}$  for refinement as:

$$I^j = \arg \min_{I^i \in \mathcal{I}} |\mathcal{P}(f) - \tau(f)|^2 \tag{8}$$

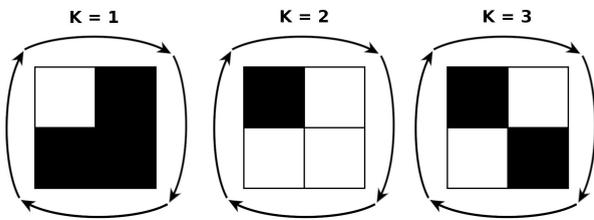
where  $\mathcal{P}(f)$  is the projected area of the marker  $f$  in the image  $I^i$  and  $\tau(f)$  the optimum marker length for refinement. Please notice that in order to refine the corners, there must be a minimum separation of  $s_{min}$  pixels between them. Thus, we define  $\tau(f) = s_{min} \times s(f)$ . For instance, if  $s_{min} = 10$ , for a marker  $f$  such that  $s(f) = 12$ , then we have that  $\tau(f) = 120$ . Finally, let us point out that if a marker looks very small in the original image  $I^0$  (i.e.,  $\mathcal{P}(f) < \tau(f)$ ), its corners are not refined neither used for pose estimation.

Fig. 8 shows the result of the proposed method. In Fig. 8a we show an input image where the two internal markers (shown in green) have been detected using the method described in Section IV-A. Fig. 8b shows the projected inner corners after the first pose estimation. Finally, Fig. 8c shows in red the refined corner locations with the proposed method. As can be observed, the initially projected corners (green) are not as precisely located as the refined ones. The refined corners are employed later to obtain a more precise estimation of the marker pose.

The corner refinement process must also consider the possibility of occlusion, i.e., the refinement process cannot be done for markers that are occluded in the image. In order to account for that possibility, a couple of conditions are analyzed for each corner during the refinement process. First, it is analyzed if the region around the corner has low contrast. Since we are dealing with black and white markers, we can expect a corner to be in a region of high contrast, thus, if the difference between the brightest and darkest pixels within the corner region is smaller than  $\tau_c$ , the corner is considered occluded and discarded from the process. Second, we discard corners that undergo large displacements during the refinement process.



**FIGURE 9.** (a) Original image showing the region of interest. (b) Results of applying the FAST detector (blue dots). (c) Examples of corner classification (d) Filtered and classified keypoints. Each color (blue, green and red dots) represent a different keypoint class.



**FIGURE 10.** The three categories a keypoint can belong to. Each keypoint will be assigned to one of these three categories, or discarded.

**D. KEYPOINT-BASED MARKER DETECTION**

In case that after the *marker detection* step (Section IV-A) no marker has been detected, our method aims at finding the Fractal Marker using the previously available detection. To do so, our method searches for the marker corners around their last observed location using a keypoint-based approach that can be enunciated as follows.

**1) REGION OF INTEREST ESTIMATION**

If the movement of the marker (or the camera) is not very fast, the marker should appear in the next frame near to its location in the previous one. In order to speed up the process, a region of interest is defined to limit the area for corner detection (next step). The region is defined around the center of the previous marker detection, with an area slightly larger than the previously observed marker area (Fig. 9a). Indeed, in case of large camera movements between frames, the region of interest may not cover the new marker position and thus the marker may not be found. In that case, it will be necessary to wait until a marker is detected using the previously explained method (Section IV-A).

**2) CORNER DETECTION AND CLASSIFICATION**

The FAST keypoint detection algorithm [34] is applied in the region of interest (Fig. 9b) and a couple of controls are established for each detected keypoint in order to remove these unlikely to belong to marker corners. First, keypoints with a low response of the FAST detector are removed, retaining only these above the 20th percentile. Second, a keypoint is

removed if the contrast in a squared neighborhood region of  $l \times l$  pixels, is below  $\tau_c$ . We have experimentally observed that the value  $l = 10$  provides good results. For the remaining keypoints, we apply a novel algorithm that analyzes if it belongs to one of the three possible categories  $K \in \{1, 2, 3\}$  shown in Fig. 10. Please notice, that these are the three types of corners that a marker can have. It can be seen as a very simple keypoint descriptor with only three different values.

The proposed method for keypoint classification is explained in Algorithm 1. First, the region around the keypoint is binarized using the average pixel intensity as threshold. Then, connected components are computed and the simple rules shown in lines 5-13 are applied for classification. The classification result of keypoints in Fig. 9b is shown in Fig. 9(c-d), where the keypoint  $K = 1$  are shown in green color,  $K = 2$  in red color and  $K = 3$  in blue color.

**Algorithm 1** Keypoint Classification

```

1:  $R \leftarrow \text{roi}(I, k, l)$  # Region of interest for image  $I$ , centered
   in the keypoint  $k$  with region size  $l \times l$ 
2:  $R^b \leftarrow \text{thresholdAvg}(R)$  # Binarize  $R$  using the average
   pixel intensity as threshold
3:  $C \leftarrow \text{connectedComponents}(R^b)$  # Determine the num-
   ber of connected components of  $R^b$ 
4:  $K \leftarrow 0$  # Init class
5: if  $C = 2$  then
6:   if  $\text{countNonZero}(R^b) > \text{countZero}(R^b)$  then
7:      $K \leftarrow 1$ ; # Set  $k$  as class 1
8:   else
9:      $K \leftarrow 2$ ; # Set  $k$  as class 2
10:  end if
11: else if  $C > 2$  then
12:    $K \leftarrow 3$ ; # Set  $k$  as class 3
13: end if
14: return  $K$ 

```

**3) RANSAC KEYPOINT MATCHING**

Once the keypoints have been classified, the next step consists in determining to which internal marker corner ( $W^i$ ) each keypoint corresponds to. Although the classification helps to

drastically reduce the number of candidates, it is not enough to uniquely match it. Using the previous Fractal Marker detection, it is possible to reduce even further the possible matches by setting a radius search  $r$ , which is automatically calculated based on the visible area occupied by the marker. Assuming that the camera/marker movement is not very large, the detected keypoint must correspond to any of the inner corners observed within the search region in the previous image. Even so, more than one inner corner of the same class can be assigned to each keypoint. Thus, a method to robustly match each keypoint to its corresponding inner corner is proposed using a RANSAC approach.

The basic idea is that there exists a homography that relates the inner corners  $W^i$  to the observed keypoints in the camera image. The minimum number of correspondences to compute such homography is four, and if the correspondences are correct, then, the homography will project the inner corner very near to a detected keypoint of the appropriate class. In that case, we have an inlier, and if the homography computed using these four points is good, then, it must produce a lot of inliers. Using these ideas, a RANSAC algorithm is employed to compute the correspondences. The algorithm will stop when a maximum number of iterations ( $n_{it}$ ) is reached, or if the percentage of inliers is above a percentage of the total number of inner corners  $\alpha$ . If the maximum number of iterations is reached, the obtained solution is considered valid if the number of inliers is greater than a percentage  $\beta$ .

As a result of the previous steps, an initial set of inner marker corners is obtained that is used to obtain an initial camera pose. The reader is referred to the Fig. 6, where the general workflow is explained.

## V. EXPERIMENTS AND RESULTS

This section explains the experiments conducted to validate our proposal. A total of five experiments have been carried out in order to compare the performance of the proposed Fractal Markers versus traditional markers. Our experiments aims at evaluating the range detection capability, the robustness to partial occlusion, the precision in the estimation of the pose and the speed of the proposed method. For comparison, the ArUco library [4] has been used as the traditional markers system.

The experiments have been performed using an iPhone SE using an image resolution of  $3840 \times 2160$  and all the images and videos employed for experiments are publicly available.<sup>1</sup> The experiments have been conducted using a single CPU of an Intel®Core™ i7-7500U 2.70GHz x 4-core processor with 8GB RAM running Ubuntu 18.04. The values for the parameters of our method employed in the tests are shown in Table 1.

### A. DETECTION RANGE ANALYSIS

This experiment aims at comparing the detection ranges of the proposed method with traditional markers. We have printed a

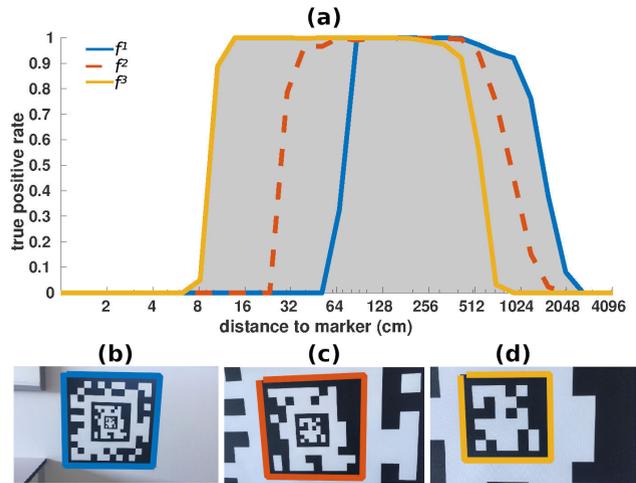
TABLE 1. Parameters values used in our experimentation.

Parameter	Value	Description
$m$	3	Number of internal markers of Fractal Marker $F$
$s(f^1)$	14	Length of black region of internal marker $f^1$ of Fractal Marker $F$
$n(f^1)$	12	Length of identification region of internal marker $f^1$ of Fractal Marker $F$
$k(f^1)$	6	Length of white region of internal marker $f^1$ of Fractal Marker $F$
$s(f^2)$	12	Length of black region of internal marker $f^2$ of Fractal Marker $F$
$n(f^2)$	10	Length of identification region of internal marker $f^2$ of Fractal Marker $F$
$k(f^2)$	4	Length of white region of internal marker $f^2$ of Fractal Marker $F$
$s(f^3)$	8	Length of black region of internal marker $f^3$ of Fractal Marker $F$
$n(f^3)$	6	Length of identification region of internal marker $f^3$ of Fractal Marker $F$
$k(f^3)$	0	Length of white region of internal marker $f^3$ of Fractal Marker $F$
$s_{min}$	10	Optimal spacing between bits used in the refinement process (Section IV-C).
$l$	10	Region size used to classify corners according to the three possible categories (Section IV-D).
$n_{it}$	500	Maximum number of iterations used by Ransac (Section IV-D).
$\alpha$	0.7	Percentage of matches needed to consider a generated model as good (Section IV-D).
$\beta$	0.1	Percentage of minimum matches necessary to consider the model as a candidate (Section IV-D).
$\tau_c$	25	Contrast threshold for corners (Sections IV-C and IV-D).

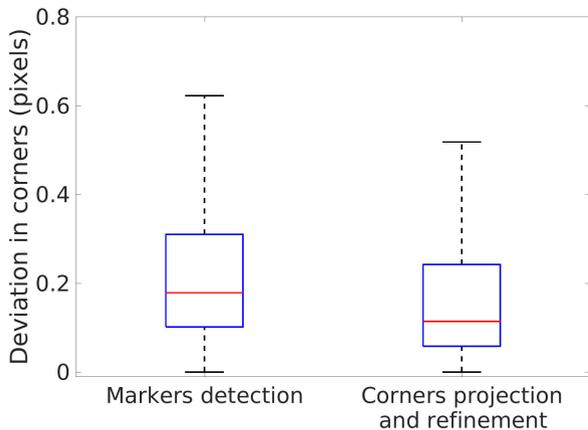
Fractal Marker comprised of three internal markers  $f^1, f^2, f^3$  with side lengths of 41.3 cm, 17.5 cm and 5.9 cm, respectively. Five video sequences (a total of 10445 frames) have been recorded starting from a very distant location from the marker (so that it can not be detected) and approaching to the marker until the camera autofocus is no longer able to obtain a clear image. Fig. 11(b-d) show images from one of the video sequences at different distances. The colored lines enclosing the markers (blue, red and yellow) have been overlaid on the images to ease the explanation of the figure.

The video sequences have been processed using both our method and the ArUco library. For that purpose, ArUco has been appropriately adapted to detect the inner markers of the Fractal Marker by ignoring the bits in the central region of side length  $k(f^i)$ . In this way, we can compare the results of ArUco and our method in the same video sequence (and thus the same conditions). Fig. 11a shows the True Positive Rate (TPR) of both methods as a function of the distance to the marker. While the colored lines show the TPR for each individual marker using ArUco, the grey area corresponds to our fractal approach. Please notice that the horizontal axis is in logarithmic scale. As can be observed, the proposed Fractal Marker can be detected within a large range of distances, i.e. [7, 2000] cm, while each individual marker has a much more reduced detection range.

<sup>1</sup><https://mega.nz/#F!qyA1QAhr!BqwdzE-tqJI2BrbzDZRcag>



**FIGURE 11.** (a) True positive detection rates as a function of the distance to the markers. Each coloured line correspond to one of the inner markers that compose the Fractal Marker. The grey area correspond to the detection range of the complete Fractal Marker. (b-d) Different views of the Fractal Marker employed for the experiments.



**FIGURE 12.** Vertex jitter before and after the proposed corner refinement. The proposed method improves accuracy.

**B. VERTEX JITTER ANALYSIS**

Vertex jitter refers to the standard deviation in the estimation of the corners that a method obtains in a sequence of images where neither the marker nor the camera moves. The standard deviation from the central position is an indication of the method precision. Please notice that error in the corners estimation is propagated to the pose (Eq. 7). This experiment aims at analyzing the impact of the proposed method for corner projection and refinement (Section IV-C) in the vertex jitter. A total of seven video sequences have been recorded pointing at a Fractal Marker (with three inner markers of side lengths 15 cm, 6.4 cm and 2.1 cm) at different distances between 49 cm and 2.74 m, having both the camera and the marker static.

Fig. 12 shows the vertex jitter of the original ArUco marker detection method (i.e., the output of Markers Detection (see Fig. 6), and after applying the whole proposed workflow (i.e., after Corner projection and refinement). As can be

**TABLE 2.** Average Computing times (in milliseconds) of the different steps involved in Fractal Marker detection and tracking.

Process	Avg (ms)
Markers detection (ArUco)	29.6
Keypoint-based marker detection	6.9
Camera pose estimation (x2)	0.2
Corners projection and refinement	1.0

observed, the proposed method for corner refinement allows reducing the vertex jitter. As a consequence, a more stable and precise camera pose estimation can be expected.

**C. COMPUTING TIMES**

The goal of this section is to show the computing times of each one of the components of our system. Indeed, our method requires more computing time than a system that detect only markers, since we perform a series of additional steps. Table 2 shows the average computing times employed by the different step shown in Fig. 6 using a total of 1037 images of resolution 3840 × 2160. For our tests, ArUco [4] library has been used for marker detection using the *DM\_NORMAL* mode.

As can be seen, the steps proposed in this work adds relatively small overload to the total computing time. The initial step “Marker Detection”, which is the same as in traditional marker detection, is the most time-consuming process. It must be remarked, though, that the number of internal markers of the Fractal Marker has no meaningful impact on the computing time of this step. Also, please notice that the “Keypoint-based marker detection” process is only necessary when none of the internal markers are detected in the first step. Thus, in most of the cases, our method will only add a negligible amount of time to the total computation.

**D. FRACTAL MARKER DETECTION WITH OCCLUSION**

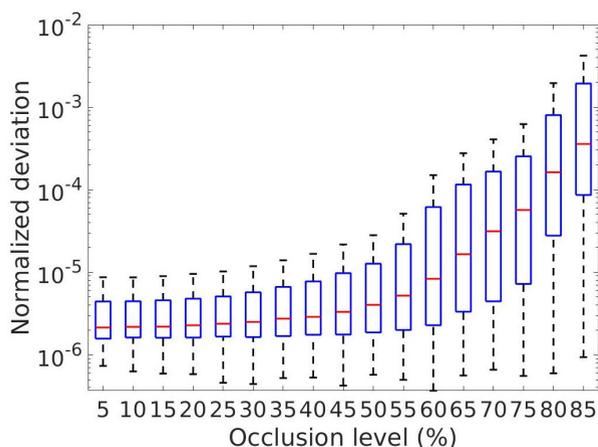
The goal of the following experiment is to analyze the robustness and precision of the proposed method in detecting Fractal Markers under several degrees of occlusion. Please notice that the tracking capabilities of our method are not tested in this experiment but in the next Section.

A total of 60 images have been taken showing three different Fractal Markers from different viewpoints and distances (ranging from 10 cm to 1.5 m) under controlled indoor illumination. The first Fractal Marker has two inner markers of side lengths 29.0 cm and 7.2 cm, the second Fractal Marker has three inner markers of side lengths 29.0 cm, 11.5 cm and 2.9 cm, and the third Fractal Marker has four inner markers of side lengths 29.0 cm, 14.5 cm, 3.6 cm and 0.9 cm.

To produce systematic occlusion, [39] proposes the use of a white paper template on the marker located in the bottom corner of the marker so that the surface of the marker was gradually overlapped. In our experiments, to know exactly the percentage of the occluded area, circles of random radius have been overlaid at random locations into the marker, as shown in Fig. 13. The color of a circle is randomly selected



**FIGURE 13.** Some of the images employed to test detection under occlusion. Different levels of occlusion are synthetically added to the images: (a) 11.29%, (b) 33.19%, (c) 53.92%, (d) 73.37%.



**FIGURE 14.** Average (red) and Standard deviation (blue) of the normalized error for different occlusion levels. See text for details.

as white or black. Since it is a synthetic occlusion, we know exactly the percentage of the marker that is occluded. For each marker, we have generated a total of 1000 synthetic images (3000 in total), so that the resulting occlusion levels are equally distributed in the range [1, 85]%. Above 85% detection becomes almost impossible.

The ground truth of an image are the locations of the four most external corners of  $f^1$  obtained without occlusion. Then, for each image with occlusion, the error is measured as the average distance between the ground truth locations and the estimated using our method. Please notice that the distance is measured in pixels, and thus the error is inversely proportional to the distance to the marker (or to the area occupied by the marker in the image). In order to correct this effect and being able to compare the results of images taken at different distances, the error is normalized dividing by the area of the marker in the image.

The results obtained are shown in Fig. 14 as box plots (average and standard deviation). The results obtained show that when the occlusion level is below 50%, it has a negligible impact on the error. For larger values of occlusion, the precision starts to be affected. In contrast to traditional marker detectors such as ArUco or AprilTag that are not robust to occlusion, our method exhibits a very robust behavior.

### E. ANALYSIS OF KEYPOINT-BASED MARKER DETECTION

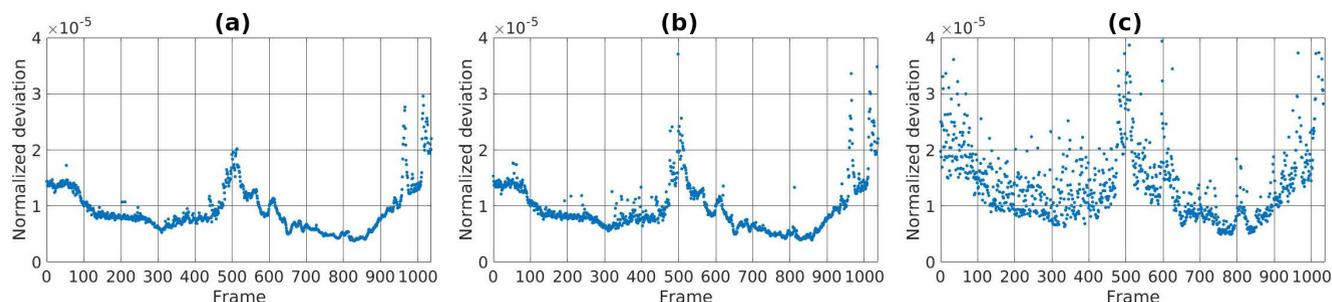
Our proposal includes a method to detect a Fractal Marker even when no internal markers have been detected. Our proposal for detection in these situations relies on a novel type of keypoint descriptor combined with the RANSAC algorithm. This section aims at analyzing the precision and robustness of the Keypoint-based marker detection. To do so, we have employed a video sequence of 1037 frames where a Fractal Marker composed by three inner markers of side lengths 15 cm, 6.4 cm and 2.1 cm was recorded at different distances (ranging from 28 cm to 1.44 m) and under controlled indoor illumination.

If we process the video sequence using the proposed workflow (Fig. 6), the keypoint-based marker detector would never be applied since at least one marker is detected in every frame. In order to be able to analyze the Keypoint-based marker detection, we force the system to follow that path, i.e., assuming that no markers have been detected except for the first frame.

The ground truth of each frame consists in the four corners of the most external marker of the Fractal Marker, computed with our method using the regular workflow. Then, the result is compared to the location estimated following the Keypoint-based marker detection path, and the error normalized dividing by the marker area observed in the frame. The results are shown in Fig. 15a. The highest values are observed around frame 800 because the camera is nearer to the camera. Nevertheless, it can be observed that the differences with the standard method are negligible.

The impact of occlusion in the error has been analyzed by synthetically adding it as in the previous experiment. For each frame, random circles have been drawn on the marker, simulating occlusions of 30% and 60%. A total of 20 synthetic images were used for each frame and occlusion percentage. The average errors obtained are shown in Fig. 15(b-c). As can be seen, the errors for a 30% occlusion are similar to these when there is no occlusion. Nevertheless, for occlusion of 60%, we can see an increase in the error.

As a conclusion, we can indicate that the proposed method for Fractal Marker Detection is reliable under occlusion.



**FIGURE 15.** Normalized pixel error of the Keypoint-based marker detection method for one video sequence using different levels of synthetic occlusion: (a) 0%, (b) 30%, (c) 60%.

## VI. CONCLUSION

This paper has proposed the Fractal Marker, a novel type of marker that can be detected in a wider range of distances than traditional fiducial markers. Fractal Markers are comprised of a set of rectangular markers, one into another, in a recursive manner. We propose a method to design Fractal Markers with an arbitrary number of inner markers so that its detection range can be increased by adding more levels.

In addition, this paper proposes a method for detecting Fractal Markers under severe occlusions. In contrast to traditional markers that are very sensitive to occlusion, our method can detect highly occluded markers at a minimum computing cost. Even if no markers can be detected in the first stage of the process, our proposed method is capable of detecting the marker by a novel keypoint-based method. We propose a very basic type of keypoint that distinguishes the three type of corners that a marker can have and a novel RANSAC-based algorithm to detect the Fractal Marker based on these keypoints.

The experiments conducted show that the proposed method is reliable and accurate, adding little computation time to the traditional marker detection step. Finally, we would like to indicate that the proposed method has been integrated as part of the ArUco library,<sup>2</sup> and is publicly available for other researchers to use it.

As possible future work, we point out the possibility of generating multiple Fractal Markers for those applications that need more than one.

## REFERENCES

- [1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Jun. 2017.
- [2] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.
- [3] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1317–1324, Jul. 2010.
- [4] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, 2014.
- [5] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 3400–3407.
- [6] F.-E. Ababsa and M. Malle, "Robust camera pose estimation using 2D fiducials tracking for real-time augmented reality systems," in *Proc. ACM SIGGRAPH Int. Conf. Virtual Reality Continuum Appl. Ind. (VRCAI)*, 2004, pp. 431–435.
- [7] V. Mondéjar-Guerra, S. Garrido-Jurado, R. Muñoz-Salinas, M.-J. Marín-Jiménez, and R. Medina-Carnicer, "Robust identification of fiducial markers in challenging conditions," *Expert Syst. Appl.*, vol. 93, no. 1, pp. 336–345, 2018.
- [8] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image Vis. Comput.*, vol. 76, pp. 38–47, Aug. 2018.
- [9] M. Koeda, D. Yano, N. Shintaku, K. Onishi, and H. Noborio, "Development of wireless surgical knife attachment with proximity indicators using ArUco marker," in *Human-Computer Interaction. Interaction in Context*, M. Kurosu, Ed. Cham, Switzerland: Springer, 2018, pp. 14–26.
- [10] S. Pflugi, R. Vasireddy, T. Lerch, T. M. Ecker, M. Tannast, N. Boemke, K. Siebenrock, and G. Zheng, "Augmented marker tracking for periacetabular osteotomy surgery," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 13, no. 2, pp. 291–304, Feb. 2018.
- [11] H. Choi, Y. Park, S. Lee, H. Ha, S. Kim, H. S. Cho, and J. Hong, "A portable surgical navigation device to display resection planes for bone tumor surgery," *Minimally Invasive Therapy Allied Technol.*, vol. 26, no. 3, pp. 144–150, 2017.
- [12] X. Zhang, M. Li, J. H. Lim, Y. Weng, Y. W. D. Tay, H. Pham, and Q.-C. Pham, "Large-scale 3D printing by a team of mobile robots," *Autom. Construct.*, vol. 95, pp. 98–106, Nov. 2018.
- [13] R. Caccavale, M. Saveriano, A. Finzi, and D. Lee, "Kinesthetic teaching and attentional supervision of structured tasks in human-robot interaction," *Auto. Robots*, vol. 43, pp. 1291–1307, Aug. 2019.
- [14] M. Bhargavapuri, A. K. Shastri, H. Sinha, S. R. Sahoo, and M. Kothari, "Vision-based autonomous tracking and landing of a fully-actuated rotorcraft," *Control Eng. Pract.*, vol. 89, pp. 113–129, Aug. 2019.
- [15] J. Bacik, F. Durovsky, P. Fedor, and D. Perdukova, "Autonomous flying with quadcopter using fuzzy control and ArUco markers," *Intell. Service Robot.*, vol. 10, pp. 185–194, Jul. 2017.
- [16] H. Sarmadi, R. Muñoz-Salinas, M. Á. Berbís, A. Luna, and R. Medina-Carnicer, "3D reconstruction and alignment by consumer RGB-D sensors and fiducial planar markers for patient positioning in radiation therapy," *Comput. Methods Programs Biomed.*, vol. 180, Oct. 2019, Art. no. 105004.
- [17] G. Alarcón-Nieto, J. M. Graving, J. A. Klarevas-Irby, A. A. Maldonado-Chaparro, I. Mueller, and D. R. Farine, "An automated barcode tracking system for behavioural studies in birds," *Methods Ecol. Evol.*, vol. 9, no. 6, pp. 1536–1547, 2018.
- [18] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin, "Dazed: Measuring the cognitive load of solving technical interview problems at the whiteboard," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng., New Ideas Emerg. Technol. Results (ICSE-NIER)*, May 2018, pp. 93–96.
- [19] R. Muñoz-Salinas, H. Sarmadi, D. Cazzato, and R. Medina-Carnicer, "Flexible body scanning without template models," *Signal Process.*, vol. 154, pp. 350–362, Jan. 2019.

<sup>2</sup><https://www.uco.es/investiga/grupos/ava/node/68>

- [20] C. Castillo, V. Marín-Moreno, R. Pérez, R. Muñoz-Salinas, and E. Taguas, "Accurate automated assessment of gully cross-section geometry using the photogrammetric interface FreeXSapp," *Earth Surf. Processes Landforms*, vol. 43, no. 8, pp. 1726–1736, 2018.
- [21] V. Babin, D. St-Onge, and C. Gosselin, "Stable and repeatable grasping of flat objects on hard surfaces using passive and epicyclic mechanisms," *Robot. Comput.-Integr. Manuf.*, vol. 55, pp. 1–10, Feb. 2019.
- [22] J. Čejka, F. Bruno, D. Skarlatos, and F. Liarokapis, "Detecting square markers in underwater environments," *Remote Sens.*, vol. 11, no. 4, p. 459, 2019.
- [23] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate  $O(n)$  solution to the PnP problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2009.
- [24] G. Bradski and A. Kaehler, *Learning OpenCV 3: Computer Vision in C++ With the OpenCV Library*, 2nd ed. Newton, MA, USA: O'Reilly Media, 2013.
- [25] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. 2nd IEEE ACM Int. Workshop Augmented Reality (IWAR)*, 1999, pp. 85–94.
- [26] D. Wagner and D. Schmalstieg, "ARToolKitPlus for pose tracking on mobile devices," in *Proc. Comput. Vis. Winter Workshop*, Jan. 2007, pp. 139–146.
- [27] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.
- [28] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L. M. Encarnação, M. Gervautz, and W. Purgathofer, "The *studierstube* augmented reality project," *Presence, Teleoper. Virtual Environ.*, vol. 11, pp. 33–54, Feb. 2002.
- [29] D. Flohr and J. Fischer, "A lightweight ID-based extension for marker tracking systems," in *Proc. Eurograph. Symp. Virtual Environ. (EGVE)*, 2007, pp. 59–64.
- [30] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and R. Medina-Carnicer, "Generation of fiducial marker dictionaries using mixed integer linear programming," *Pattern Recognit.*, vol. 51, pp. 481–491, Mar. 2016.
- [31] H. Álvarez, I. Leizea, and D. Borro, "A new marker design for a robust marker tracking system against occlusions," *Comput. Animation Virtual Worlds*, vol. 23, no. 5, pp. 503–518, 2012.
- [32] M. Krogius, A. Haggemiller, and E. Olson, "Flexible layouts for fiducial tags," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019. [Online]. Available: <https://april.eecs.umich.edu/papers/details.php?name=krogius2019iros>
- [33] H. Wang, Z. Shi, G. Lu, and Y. Zhong, "Hierarchical fiducial marker design for pose estimation in large-scale scenarios," *J. Field Robot.*, vol. 35, no. 6, pp. 835–849, 2018.
- [34] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," 2008, *arXiv:0810.2434*. [Online]. Available: <https://arxiv.org/abs/0810.2434>
- [35] S. Suzuki, "Topological structural analysis of digitized binary images by border following," *Comput. Vis., Graph., Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.
- [36] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geograph. Inf. Geovis.*, vol. 2, no. 10, pp. 112–122, 1973.
- [37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
- [38] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 252–286, Sep. 2014.
- [39] K. Shabalina, A. Sagitov, M. Svinin, and E. Magid, "Comparing fiducial markers performance for a task of a humanoid robot self-calibration of manipulators: A pilot experimental study," in *Proc. 3rd Int. Conf. ICR*, Leipzig, Germany, Sep. 2018, pp. 249–258.



**FRANCISCO J. ROMERO-RAMIREZ** received the bachelor's degree in computer science and the master's degree in geomatics and remote sensing from the Universidad de Córdoba, Spain, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Departamento de Informática y Análisis Numérico. Until 2017, he was a member of the Department of Forestry Engineering, Universidad de Córdoba. He has participated in several national and European research projects where his main tasks have been mainly focused on the processing and analysis of multispectral images and LiDAR. His current research interest includes computer vision and location.



**RAFAEL MUÑOZ-SALINAS** received the bachelor's degree in computer science and the Ph.D. degree from the University of Granada, Spain, in 2003 and 2006, respectively. Since 2006, he has been with the Department of Computing and Numerical Analysis, Universidad de Córdoba, where he is currently a Full Professor. He is also a Researcher with the Biomedical Research Institute of Cordoba (IMIBIC). His current research interests include computer vision, soft computing techniques applied to robotics, and human-robot interaction. He has coauthored more than 100 articles in conferences, books, and top-ranked journals. One of his articles was the most-cited of the prestigious journal Pattern Recognition and is considered a highly-cited article. He has supervised seven Ph.D. students and participated in more than 20 projects, both industrial and scientific. He has been a Visiting Researcher with the Universities of DeMontfort, U.K.; Örebro, Sweden; TUM, Munich; INRIA, France; Groningen, The Netherlands, and Luxembourg. As a Teacher, he has been teaching for more than 12 years, supervised more than 30 final degree projects, and taught three international courses with the Universities of Groningen, Luxembourg, and Brno. In addition, he has been a part of the Erasmus STA teaching mobility projects with the Universities of Malta, Coimbra (Portugal), and Dubrovnik (Croatia).



**R. MEDINA-CARNICER** received the bachelor's degree in mathematics from the University of Seville, Spain, and the Ph.D. degree in computer science from the Polytechnic University of Madrid, Spain, in 1992. Since 1996, he has been the Head of the Computer Vision Group AVA, Universidad de Córdoba, Spain, and a Full Professor, since 2012. He is currently a Researcher with the Biomedical Research Institute of Córdoba (IMIBIC). He has been a principal investigator of more than ten research projects. His current research interests include computer vision techniques applied to robotics, biomedicine, and augmented reality.

...