Fractal Markers: a new approach for long-range marker pose estimation under occlusion

Preprint · April 2019

DOI: 10.13140/RG.2.2.31185.17767

Francisco J. Romero-Ramirez
University of Cordoba (Spain)

Rafael Muñoz-Salinas
University of Cordoba (Spain)

Rafael Medina-Carnicer
University of Cordoba (Spain)

Fractal Markers: a new approach for long-range marker pose estimation under occlusion

Francisco J. Romero-Ramirez[1,*], Rafael Muñoz-Salinas[1,2,], Rafael Medina-Carnicer[1,2,]

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.

Keywords: Fiducial Markers, Marker Mapping, Pose Estimation

1. Introduction

Pose estimation is a common problem in many applications requiring a precise localization such as autonomous robots or augmented reality applications. Artificial fiducial markers have recently become an attractive method to solve this problem. These markers are designed to be easily detected in the images allowing a real-time estimation of the location. Among the detection systems based on fiducial markers, the square-based ones are the most popular, consisting of an external black border and an internal code (most often binary) that uniquely identifies each marker. Using its four outer vertices it is possible to estimate the pose with respect to the marker [1, 2, 3, 4].

Using fiducial markers for pose estimation have two main limitations. First, due to the fixed size of the marker when printed, 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 Fig. 1(a). The second problem is the *occlusion problem.* Most marker detection methods are incapable of dealing with occlusions (Fig. 1(c)), and those that deal with it, are very slow.

This paper proposes a novel type of marker, the *Fractal Marker*, designed as the composition of squared fiducial markers of different sizes, one into another. As shown in Fig. 1, the proposed fractal marker (Fig. 1b) can be detected from a wider range of distances than a single marker (Fig. 1a). Also, it alleviates the partial occlusion problem, since the pose can be estimated from any marker even if the most external one is occluded. 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. In order to do so, we propose a novel type of keypoint descriptor, which distinguishes the three type of corners that can be found in a squared fiducial marker. As a consequence, our method is not only capable of detecting the marker in case of occlusion (Fig. 1(d)), 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 a high robustness to occlusion, while adding little computational cost. The proposed method is ideal in certain applications where only partial view of the marker is expected, or it must be detected from a wide range of distances (e.g. augmented reality applications or drone landing).

The remainder of this work is organized as follow. Section 2 reviews the related works, while Section 3 explains the design of Fractal markers and Section 4 describes the proposed method for pose estimation using them. Finally, Section 5 shows the experimentation carried out and Sec-

[*] Corresponding author

Email addresses: fj.romero@uco.es (Francisco J. Romero-Ramirez), in1musar@uco.es (Rafael Muñoz-Salinas), rmedina@uco.es (Rafael Medina-Carnicer)

[1] Departamento de Informática y Análisis Numérico, Edificio Einstein. Campus de Rabanales, Universidad de Coŕdoba, 14071, Córdoba, Spain, Tlfn:(+34)957212289

[2] Instituto Maimónides de Investigación en Biomedicina (IMIBIC). Avenida Menéndez Pidal s/n, 14004, Córdoba, Spain, Tlfn:(+34)957213861

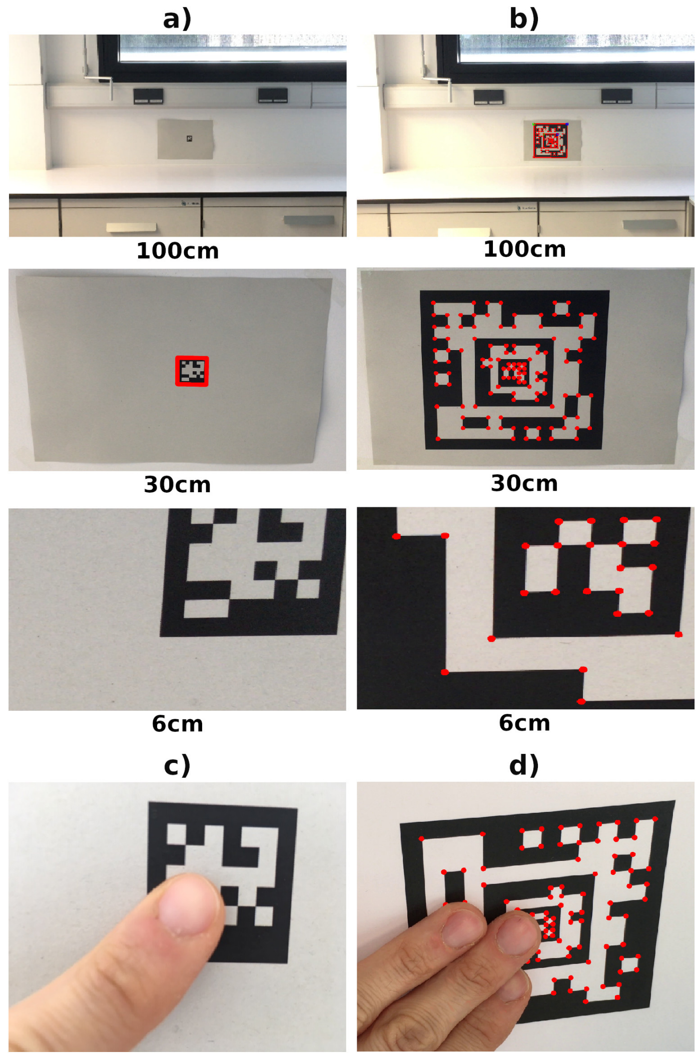

Figure 1: Left column shows the results of ArUco [1]. while the right column is our proposal. The images show the typical problems of squared markers: the resolution and occlusion problems. Our new type of marker, the Fractal marker, alleviates these problems.

tion 6 draws some conclusions.

Finally, we would like to indicate that the proposed method has been integrated as part of the ArUco library, and is publicly available [3] for other researchers to use it.

2. Related Works

As previously indicated, fiducial markers are a very popular method for pose estimation. ARToolKit [5] 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 [6] tries to solve its deficiencies by employing a binary BCH code [7] that provides a robust detection and correction. Nevertheless, the project was finally halted and followed by Studierstube project [8].

BinARyID [9] 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 [2] which introduces methods for correction. ArUco [1] 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 [10].

A recent work [11] 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 [12] 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.* [13]. 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 [14], 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 [15] is the work most the related to ours. The authors propose a new hierarchical marker structure.

[3] `https://www.uco.es/investiga/grupos/ava/node/68`

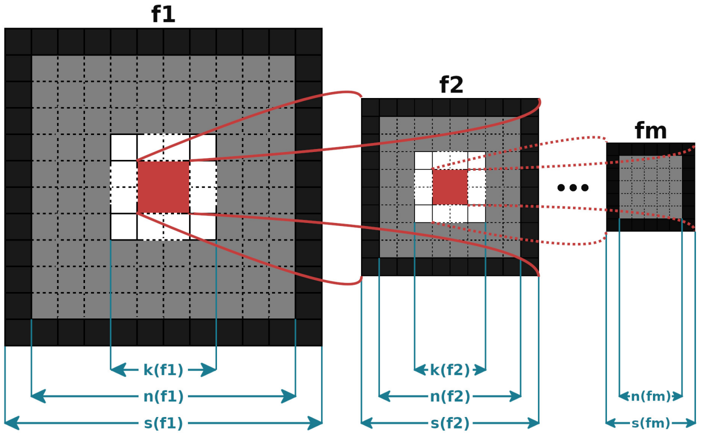

Figure 2: Generic structure of fractal marker F, in which each marker is composed of a set of cells that can be grouped into three categories. The black band correspond to the marker border, the gray cells configure and uniquely determine the marker, and finally, the white band facilitate the detection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

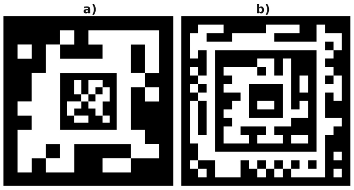

Figure 3: Example of different configurations of fractal markers and area of identification region $S_R(f^i)$. See text for details.

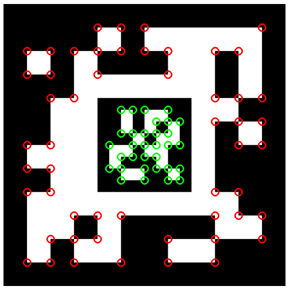

Figure 4: Fractal marker composed of two internal markers. The inner corners of marker f^1 and f^2 are shown in red and in green respectively.

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.

3. Fractal marker design

Let us define a fractal marker F as a set of m squared markers (f^1, f^2, ..., 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 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 lengths 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.

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):

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

The information bits of the codification region are randomly selected using a uniform probability distribution so that the total number of black and white bits is balanced.

Fig. 3 shows two different possible combinations of internal markers for a fractal marker. Figure. 3(a) 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. 3(b), the fractal marker is composed of three internal markers $s(f^1) = 20$, $n(f^1) = 18$, $k(f^1) = 14$, $S_R(f^1) = 128$; $s(f^2) = 12$, $n(f^2) = 10$, $k(f^2) = 6$, $S_R(f^2) = 64$ and $s(f^3) = 4$, $n(f^3) = 2$, $k(f^3) = 0$, $S_R(f^3) = 4$.

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.

A marker is considered as valid, when the Hamming distance in its four orientations is greater than zero, i.e. $H(f^i, R_j(f^i)) > 0, j \in \{90, 180, 270\}$ degrees, where H is

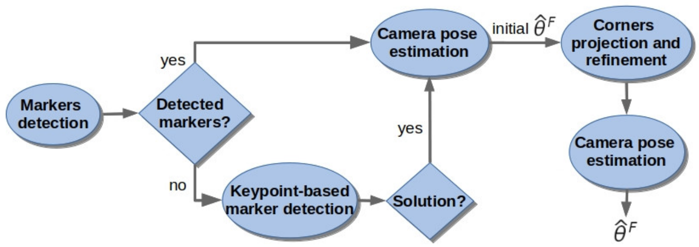

Figure 5: General workflow of proposed method for marker pose estimation.

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. Thus, we can ensure correct identification of the marker independently of its orientation.

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:

$$\begin{array}{lcl} c_1^i & = & (\ \ s(f^i)/2, \ -s(f^i)/2, \ 0\ ) \\ c_2^i & = & (\ \ s(f^i)/2, \ \ s(f^i)/2, \ 0\ ) \\ c_3^i & = & (\ -s(f^i)/2, \ \ s(f^i)/2, \ 0\ ) \\ c_4^i & = & (\ -s(f^i)/2, \ -s(f^i)/2, \ 0\ ) \end{array} \tag{2}$$

We are assumming 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. 2) of each marker f^i, there is a set of internal corners (see Figure 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 $f^i \in F$:

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

where w_j^i are the three-dimentional coordinates as w.r.t. the marker center. Figure 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

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

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

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

to the set of all the marker corners of a fractal maker F.

4. Fractal marker detection

This section explains the proposed method for detecting and tracking fractal markers under occlusion. Figure 5 depicts the workflow of our method. The first step of the process is to detect markers (Section 4.1). If at least one marker is detected, the detected corners are used to obtain an initial estimation of the marker pose (Section 4.2), 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 and then used to compute again the marker pose, which now contains more points and thus obtains a more precise location (Section 4.3).

If no makers 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[16] corner detector is employed to extract all the relevant corners in the image. The corners are then classified into the three categories previously explained. 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 4.4).

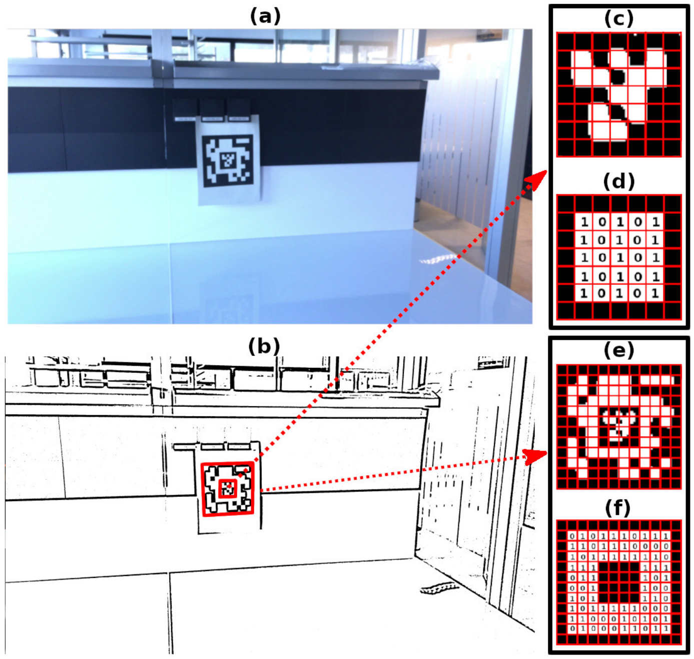

Figure 6: Detection and identification of fractal markers. (a) Original image. (b) Thresholded image showing the result of contour extraction and filtering. (c and e) Canonical images of rectangular contours containing our markers. (d and f) Binarized versions of the canonical images.

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

4.1. 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

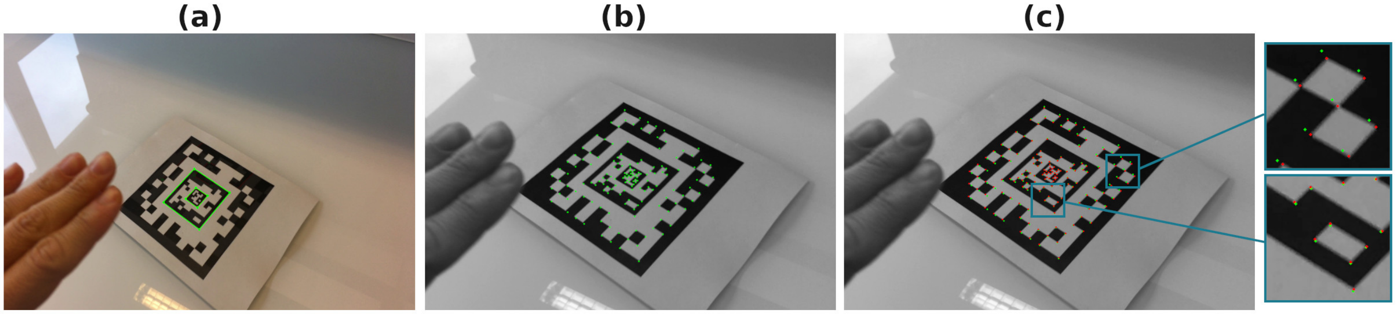

Figure 7: (a) Detection of markers and external corners in original image. (b) Initial estimation of the position using external corners of the detected markers. (c) Refinement of the pose estimation: the green points represent the estimate of the previous step (b), in red the new estimation.

employed in [1] 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. 6-(b)).

2. Contour extraction and filtering: Contour extraction of each internal marker is performed by Suzuki and Abe [17] algorithm. It provides a set of contours, many of which correspond to unwanted objects. A filtering process is carried out using Douglas and Peucker algorithm [18] which selects only the ones most similar to a polygon (Fig. 6(b)).

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 and subsequently thresholded using Otsu's method [19]. 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 (Figures 6(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 4.2.

4.2. 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 $t = (t_x, t_y, t_z)$:

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

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

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:

$$u = \Psi(\delta, \theta, p), \tag{4}$$

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 $(k_1, \dots, k_n)$.

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

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{p \in \mathcal{D}} \left[\Psi(\delta, \theta, p) - O(p)\right]^2 \tag{5}$$

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)[20].

4.3. 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. 4) 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. 5.

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.

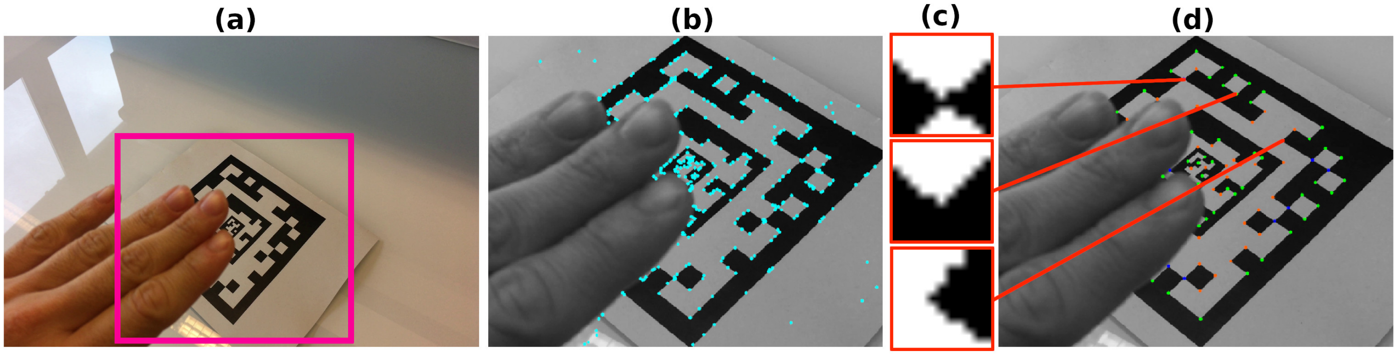

Figure 8: (a) Original image showing the region of interest. (b) Results of applying the FAST detector (blue dots). (c) Examples of corner classification (d) Filtered and classified keypoints. Each color (blue, green and red dots) represent a different keypoint class.

Let us denote $\mathcal{I} = (I^0, I^2, ..., I^p)$ 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:

$$I^j = \underset{I^i \in \mathcal{I}}{\text{argmin}} |\mathcal{P}(f) - \tau(f)^2| \tag{6}$$

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., $P(f) < \tau(f)$), its corners are not refined neither used for pose estimation.

Fig. 7 shows the result of the proposed method. In Fig. 7(a) we show an input image where the two internal markers (shown in green) have been detected using the method described in Section 4.1. Fig. 7(b) shows the projected inner corners after the first pose estimation. Finally, Fig. 7(c) 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 τ_c, the corner is considered occluded and discarded from the process. Second, we discard corners that undergo large displacements during the refinement process. We keep a list of the displacements of the corner along

4.4. KeyPoint-based marker detection

In case that after the *marker detection* step (Section 4.1) 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.

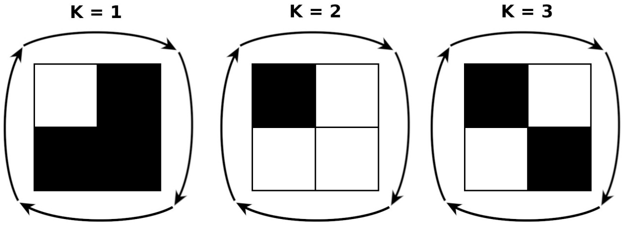

Figure 9: The three categories a keypoint can belong to. Each keypoint will be assigned to one of these three categories, or discarded.

1. *Region of interest estimation*: In order to speed up corner detection, a region of interest if defined (Fig. 8(a)) as a slightly enlarged area around the last fractal marker location. 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 previous method (Section 4.1).

2. *Corner detection and classification*: The FAST keypoint detection algorithm [16] is applied in the region of interest (Figure 8(b)) 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 τ_c. 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. 9. 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. 8(b) is shown in Figures 8(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:  $R \leftarrow roi(I, k, l)$  # Region of interest for image I, centered in the keypoint k with region size $l \times l$
2:  $R^b \leftarrow thresholdAvrg(R)$ # Binarize R using the average pixel intensity as threshold
3:  $C \leftarrow connectedComponents(R^b)$ # Determine the number of connected components of $R^b$
4:  $K \leftarrow 0$ # Init class
5:  if $C = 2$ then
6:      if $countNonZero(R^b) > countZero(R^b)$ 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 α. If the maximum number of iterations is reached, the obtained solution is considered valid if the number of inliers is greater than a percentage β.

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. 5, where the general workflow is explained.

5. 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 [1] has been used as the traditional markers system.

A set of video sequences have been recorded for the experiments with an iphone SE using a image resolution of 3840 × 2160 (they are publicly available at [4]). All experiments have been conducted using a single CPU of an Intel® Core^{TM} 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.

5.1. Detection range analysis

This experiment aims at comparing the detection ranges of the proposed method and a single traditional marker. We have printed a fractal marker of $36.5cm \times 36.5cm$, comprised by three internal marker, such that that the most internal marker is an ArUco marker of $6cm \times 6cm$. A total of eighth video sequences have been recorded starting from a very distant location from the marker, so that its area in the image so small that it can not be detected, and then the camera approaches to the marker. Figures 10(b-d) shows images from one of the video sequences.

The video sequences have been processed using both our proposed method and the ArUco library and the True Positive Rate analyzed as a function of the distance to the marker. The results are shown in Fig. 10(a). Please notice that the horizontal axis is in logarithmic scale. As can be observed, the proposed approach is able to detect the marker in a much larger range of distances. While the proposed Fractal marker can be in the range $[5, 2000]$ cm, the original ArUco is only able to detect in the range $[10, 400]$ cm.

Parameter	Value	Description
m	3	Number of internal markers of fractal marker $\mathcal{F}$
$s(f^1)$	14	Length of black region of internal marker f^1 of fractal marker $\mathcal{F}$
$n(f^1)$	12	Length of identification region of internal marker f^1 of fractal marker $\mathcal{F}$
$k(f^1)$	6	Length of white region of internal marker f^1 of fractal marker $\mathcal{F}$
$s(f^2)$	12	Length of black region of internal marker f^2 of fractal marker $\mathcal{F}$
$n(f^2)$	10	Length of identification region of internal marker f^2 of fractal marker $\mathcal{F}$
$k(f^2)$	4	Length of white region of internal marker f^2 of fractal marker $\mathcal{F}$
$s(f^3)$	8	Length of black region of internal marker f^3 of fractal marker $\mathcal{F}$
$n(f^3)$	6	Length of identification region of internal marker f^3 of fractal marker $\mathcal{F}$
$k(f^3)$	0	Length of white region of internal marker f^3 of fractal marker $\mathcal{F}$
s_{min}	10	Optimal spacing between bits used in the refinement process (Section 4.3).
l	10	Region size used to classify corners according to the three possible categories (Section 4.4).
n_{it}	500	Maximum number of iterations used by Ransac (Section 4.4).
α	0.7	Percentage of matches needed to consider a generated model as good (Section 4.4).
β	0.1	Percentage of minimum matches necessary to consider the model as a candidate (Section 4.4).
τ_c	25	Contrast threshold for corners (Sections 4.3 and 4.4).

Table 1: Parameters values used in our experimentation.

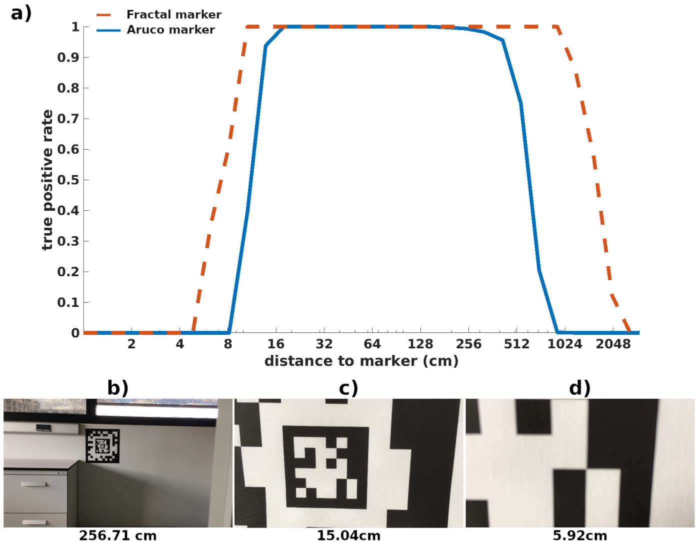

Figure 10: a) True positive detection rates of ArUco and our method as a function of the distance to the marker. View of the markers for distances: b) 256.71 cm, c) 15.04 cm, d) 5.92 cm

5.2. 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. 5). This experiment aims at analyzing the impact of the proposed method for corner projection and refinement (Section 4.3) in the vertex jitter. A total of seven video sequences have been recorded pointing at a fractal marker (with three markers) at different distances, having both the camera and the marker static.

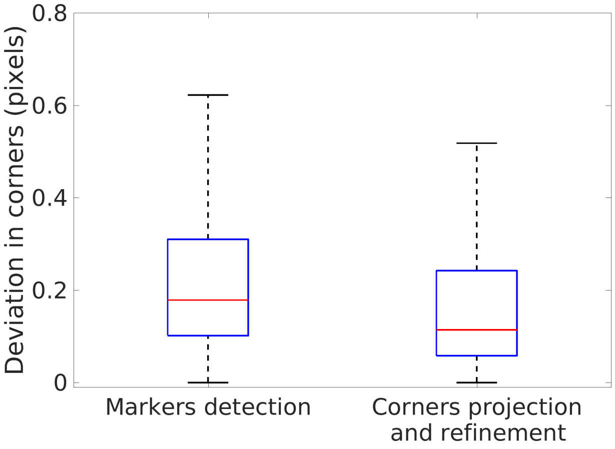

Figure 11: Vertex jitter before and after the proposed corner refinement. The proposed method improves accuracy.

Fig. 11 shows the vertex jitter of the original ArUco marker detection method (i.e., the output of Markers Detection (see Fig. 5), and after applying the whole proposed workflow (i.e., after Corner projection and refinement). As can be 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.

[4] https://mega.nz/#F!qyA1QAhR!BqwdzE-tqJI2BrbzDZRcag

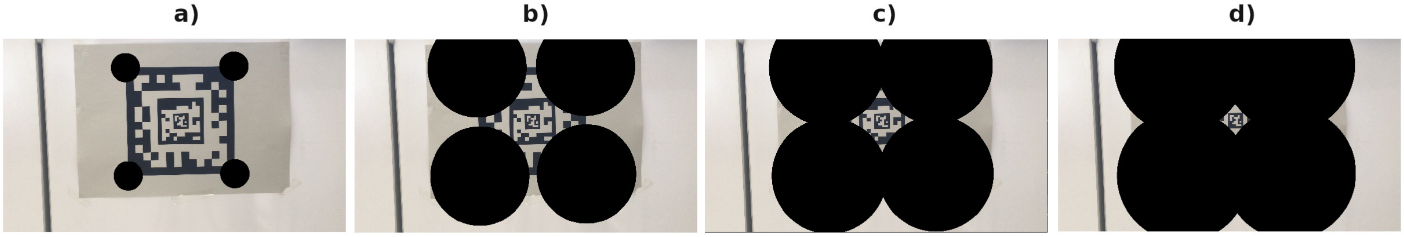

Figure 12: Some of the images employed to test detection under occlusion. Different levels of occlusion are synthetically added to the images: a) 4.33%, b) 61.25%, c) 82.95%, d) 94.51%.

5.3. 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. 5 using a total of 1037 images of resolution 3840 × 2160. For our tests, ArUco[1] 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. 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.

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

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

5.4. 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 2550 images have been taken showing a fractal marker (with three internal markers) from different viewpoints and distances. In order to simulate occlusion, a pattern consisting of four circles has been overlaid in the image. Fig. 12 shows the resulting images for different sizes of the occluding pattern. Since it is a synthetic image, we have control over the percentage of the marker that is occluded. For instance, an occlusion of 0.1% makes the external marker to be undetected, and for occlusion of 50% only the most internal marker can be detected.

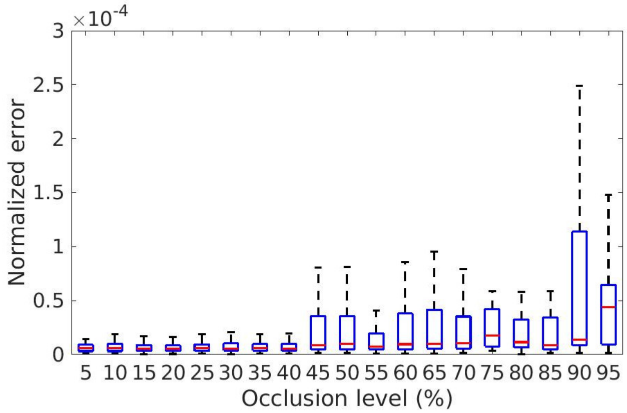

Figure 13: Average (red) and Standard deviation (blue) of the normalized error for different occlusion levels. See text for details.

The ground truth of an image are the locations of the four most external corners of f^1 obtained without occlusion. Then, for an image with artificial 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 Figure 13 as box plots (average and standard deviation). The results obtained show that when the occlusion level is below 45%, the occlusion 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.

5.5. Analysis of Keypoint-based marker detection

Our proposal includes a method to detect a Fractal Marker even when no internal markers have been detected. Examples of that situation are shown in Fig. 1(a) and

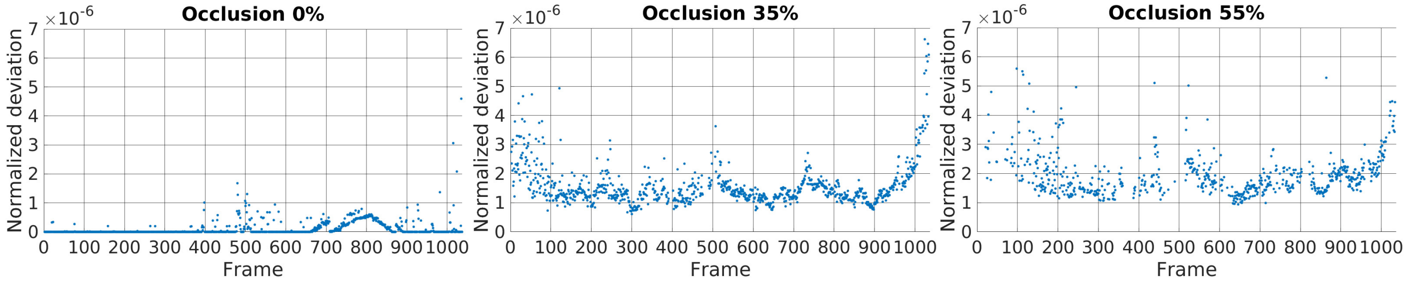

Figure 14: Normalized pixel error of the Keypoint-based marker detection method for one video sequence using different levels of synthetic occlusion.

Fig. 8(a). 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 was recorded at different distances, so that its three internal markers were detectable in all the frames.

If we process the video sequence using the proposed workflow (Fig. 5), 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 the left graph of Fig. 14. The highest values are observer 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. As in the previous case, we added synthetic occlusion to force several levels of occlusion. The results obtained for occlusions levels of 35% and 55% are shown in Fig. 14. As can be seen, the errors increase, but are still very low. As a conclusion, we can indicate that the proposed method for Fractal Marker Detection is reliable under occlusion.

6. Conclusions

This paper has proposed the Fractal Marker, a novel type of marker that can be detected in a wider range of distances that 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, and is publicly available [5] for other researchers to use it.

7. Acknowledgments

This project has been funded under projects TIN2016-75279-P and IFI16/00033 (ISCIII) of Spain Ministry of Economy, Industry and Competitiveness, and FEDER.

8. References

[1] S. Garrido-Jurado, R. Muñoz Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (6) (2014) 2280–2292.

[2] E. Olson, Apriltag: A robust and flexible visual fiducial system, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 3400–3407.

[3] F. Ababsa, M. Mallem, Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems, in: Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI '04, 2004, pp. 431–435.

[5] https://www.uco.es/investiga/grupos/ava/node/68

[4] V. Mondéjar-Guerra, S. Garrido-Jurado, R. Muñoz-Salinas, M.-J. Marín-Jiménez, R. Medina-Carnicer, Robust identification of fiducial markers in challenging conditions, Expert Systems with Applications 93 (1) (2018) 336–345.

[5] H. Kato, M. Billinghurst, Marker tracking and hmd calibration for a video-based augmented reality conferencing system, in: Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on, 1999, pp. 85–94.

[6] D. Wagner, D. Schmalstieg, ARToolKitPlus for pose tracking on mobile devices, in: Computer Vision Winter Workshop, 2007, pp. 139–146.

[7] S. Lin, D. J. Costello, Error Control Coding, Second Edition, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[8] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L. M. Encarnação, M. Gervautz, W. Purgathofer, The studierstube augmented reality project, Presence: Teleoper. Virtual Environ. 11 (1) (2002) 33–54.

[9] D. Flohr, J. Fischer, A Lightweight ID-Based Extension for Marker Tracking Systems, in: Eurographics Symposium on Virtual Environments (EGVE) Short Paper Proceedings, 2007, pp. 59–64.

[10] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using mixed integer linear programming, Pattern Recognition 51 (2016) 481–491.

[11] F. J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers, Image and Vision Computing 76 (2018) 38 – 47.

[12] M. Fiala, Designing highly reliable fiducial markers, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1317–1324.

[13] H. Álvarez, I. Leizea, D. Borro, A new marker design for a robust marker tracking system against occlusions, Comput. Animat. Virtual Worlds 23 (5) (2012) 503–518.

[14] M. Krogius, A. Haggenmiller, E. Olson, Flexible tag layouts for the AprilTag fiducial system (under review), 2019.

[15] H. Wang, Z. Shi, G. Lu, Y. Zhong, Hierarchical fiducial marker design for pose estimation in large-scale scenarios, Journal of Field Robotics 35 (6) (2018) 835–849.

[16] E. Rosten, R. Porter, T. Drummond, Faster and better: a machine learning approach to corner detection, CoRR abs/0810.2434.

[17] Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing 30 (1) (1985) 32 – 46.

[18] D. H. Douglas, T. K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartographica: The International Journal for Geographic Information and Geovisualization 2 (10) (1973) 112 – 122.

[19] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66.

[20] T. Collins, A. Bartoli, Infinitesimal plane-based pose estimation, International Journal of Computer Vision 109 (3) (2014) 252–286.