Contribuciones al uso de marcadores para Navegación Autónoma y Realidad Aumentada

Contributions to the use of fiducial markers in Autonomous Navigation and Augmented Reality

Francisco José Romero Ramírez

Directores: Rafael Muñoz Salinas

Rafael Medina Carnicer

Departamento de Informática y Análisis Numérico

Universidad de Córdoba

Programa de Doctorado: Computación Avanzada,Energía y Plasma

Mayo 2021

TITULO: *Contributions to the use of fiducial markers in Autonomous Navigation and Augmented Reality*

AUTOR: *Francisco José Romero Ramírez*

Campus de Rabanales
Ctra. Nacional IV, Km. 396 A
14071 Córdoba

https://www.uco.es/ucopress/index.php/es/
ucopress@uco.es

TÍTULO DE LA TESIS:

Contribuciones al uso de marcadores para Navegación Autónoma y Realidad Aumentada

Contributions to the use of fiducial markers in Autonomous Navigation and Augmented Reality

DOCTORANDO/A:

Francisco José Romero Ramírez

INFORME RAZONADO DEL/DE LOS DIRECTOR/ES DE LA TESIS

(se hará mención a la evolución y desarrollo de la tesis, así como a trabajos y publicaciones derivados de la misma).

La Tesis Doctoral tenía como objetivo el crear una metodología que permitiera la detección de marcadores de forma robusta minimizando el error en la estimación de la posición de la cámara, y a su vez primando el tiempo de computo de su localización, teniendo en cuenta que en la mayoría de casos estos sistemas son utilizados por dispositivos con recursos limitados de cómputo tales como móviles o vehículos aéreos.

El Plan de Investigación establecido fijaba como objetivos específicos a lograr durante el desarrollo de la Tesis:

- Metodologías que permitan de forma eficaz la detección de marcadores fiduciales, minimizando el error obtenido en la estimación de la posición.
- Contribuciones al problema de posicionamiento y seguimiento de dispositivos móviles en entornos reales.
- Contribuciones al problema de la detección de marcadores multiresolución.
- Contribuciones de técnicas que integren el uso de marcadores artifíciales con marcadores naturales para extender las capacidades de ambos.
- Aplicación de las técnicas de posicionamiento propuestas al ámbito de la navegación autónoma de vehículos aéreos y a la realidad aumentada

El doctorando ha cumplido de forma excelente el Plan de Investigación diseñado tanto en lo que se refiere al Plan de Formación establecido, que incluía tres meses de estancia internacional , como en la obtención de los resultados previstos.

Las contribuciones científicas, derivadas de la Tesis Doctoral, que han sido obtenidas por el doctorando son:

3 publicaciones JCR (Q1):

- Tracking fiducial markers with discriminative correlation filters. Romero-Ramirez, F.J., Muñoz-Salinas, R, Medina-Carnicer, R.mImage and Vision Computing (2021) Volume 107, March 2021, Article number 104094

- Fractal Markers: A New Approach for Long-Range Marker Pose Estimation under Occlusion. Romero-Ramirez, F.J., Muñoz-Salinas, R, Medina-Carnicer, R. IEEE Access (2019) Volume 7, 2019, Article number 8890613, Pages 169908-169919

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

Resaltar que el artículo “Speeded up detection of squared fiducial markers” ha recibido hasta el momento 178 citas (Scopus) and 300 citas (Google Scholar), que es el artículo más citado que ha sido publicado desde 2018 en la revista indicada y que ha recibido el "Editors Choice Award 2020".

Adicionalmente, el doctorando es co-autor de 1 publicación JCR (Q2) en colaboración con miembros del Grupo de Investigación donde se ha desarrollado la Tesis:

- 3D human pose estimation from depth maps using a deep combination of poses. Marín-Jiménez, M.J, Romero-Ramirez, F.J., Muñoz-Salinas, R, Medina-Carnicer, R. Journal of Visual Communication and Image Representation Volume 55, August 2018, Pages 627-639

Por todo ello, se autoriza la presentación de la tesis doctoral.

Córdoba, 3 de Mayo de 2021

Firma de los directores

Fdo.:_Rafael Muñoz Salinas Fdo.: Rafael Medina Carnicer

A mi familia . . .

Agradecimientos

Me gustaría en primer lugar agradecer y reconocer a mis directores de tesis Rafael Muñoz y Rafael Medina su dedicación e implicación en cada uno de los trabajos, así como por todos los conocimientos que me han aportado a lo largo de estos años.

A todo el grupo AVA y compañeros por vuestros consejos y tiempo compartido.

Agradecer a mi familia por estar siempre ahí y apoyarme en todo momento. A mi madre, padre y hermana por vuestro cariño y haberme dado la oportunidad de llegar a ser quien soy. A mis sobrinas Carlota, Lucia y Vega por la alegría con la que llenáis la casa.

A Patty por todo su apoyo y paciencia.

Abstract

Square planar markers are a widely used tools for localization and tracking due to their low cost and high performance. Many applications in Robotics, Unmanned Vehicles and Augmented Reality employ these markers for camera pose estimation with high accuracy.

Nevertheless, marker-based systems are affected by several factors that limit their performance. First, the marker detection process is a time-consuming task, which is intensified as the image size increases. As a consequence, the current high-resolution cameras has weakened the processing efficiency of traditional marker systems. Second, marker detection is affected by the presence of noise, blurring and occlusion. The movement of the camera produces image blurriness, generated even by small movements. Furthermore, the marker may be partially or completely occluded in the image, so that it is no longer detected.

This thesis deals with the above limitations, proposing novel methodologies and strategies for successful marker detection improving both the efficiency and robustness of these systems. First, a novel multi-scale approach has been developed to speed up the marker detection process. The method takes advantage of the different resolutions at which the image is represented to predict at runtime the optimal scale for detection and identification, as well as following a corner upsampling strategy necessary for an accurate pose estimation. Second, we introduce a new marker design, *Fractal Marker*, which using a novel keypoint-based method achieves detection even under severe occlusion, while allowing detection over a wider range of distance than traditional markers. Finally, we propose a new marker detection strategy based on Discriminative Correlation Filters (DCF), where the marker and its corners represented in the frequency domain perform more robust and faster detections than state-of-the-art methods, even under extreme blur conditions.

Resumen

Los marcadores planos cuadrados son una de las herramientas ampliamente utilizadas para la localización y el tracking debido a su bajo coste y su alto rendimiento. Muchas aplicaciones en Robótica, Vehículos no Tripulados y Realidad Aumentada emplean estos marcadores para estimar con alta precisión la posición de la cámara.

Sin embargo, los sistemas basados en marcadores se ven afectados por varios factores que limitan su rendimiento. En primer lugar, el proceso de detección de marcadores es una tarea que requiere mucho tiempo y este incrementa a medida que aumenta el tamaño de la imagen. En consecuencia, las actuales cámaras de alta resolución han debilitado la eficacia del procesamiento de los sistemas de marcadores tradicionales. Por otra parte, la detección de marcadores se ve afectada por la presencia de ruido, desenfoque y oclusión. El movimiento de la cámara produce desenfoque de la imagen, generado incluso por pequeños movimientos. Además, el marcador puede aparecer en la imagen parcial o completamente ocluido, dejando de ser detectado.

Esta tesis aborda las limitaciones anteriores, proponiendo metodologías y estrategias novedosas para la correcta detección de marcadores, mejorando así tanto la eficiencia como la robustez de estos sistemas. En primer lugar, se ha desarrollado un novedoso enfoque multiescala para acelerar el proceso de detección de marcadores. El método aprovecha las diferentes resoluciones en las que la imagen está representada para predecir en tiempo de ejecución la escala óptima para la detección e identificación, a la vez que sigue una estrategia de upsampling de las esquinas necesaria para estimar la pose con precisión. En segundo lugar, introducimos un nuevo diseño de marcador, Fractal Marker, que, mediante un método basado en keypoints, logra detecciones incluso en casos de oclusión extrema, al tiempo que permite

la detección en un rango de distancias más amplio que los marcadores tradicionales. Por último, proponemos una nueva estrategia de detección de marcadores basada en Discriminate Correlation Filters (DCF), donde el marcador y sus esquinas representadas en el dominio de la frecuencia realizan detecciones más robustas y rápidas que los métodos de referencia, incluso bajo condiciones extremas de emborronamiento.

Table of contents

Chapter 1

Introduction

Pose estimation is the problem of obtaining the position and orientation or an object, which is necessary for a wide range of applications. In the medical field, it is used for surgery tool tracking [1–3]; in unmanned aerial vehicles it allows to establish references for navigation, as well as assistance in landing [4, 5]; in human-machine interaction applications is applied to localization and tracking of human body parts [6, 7]; in augmented reality allows the registration of virtual objects [8, 9] (Fig.1.1).

Cameras are low-cost sensors that can be employed to efficiently solve the localization problem. In the best case, natural markers are used for localization and tracking [10–12]. However, they have limitations in poorly textured scenarios, such as corridors, clear rooms, ceilings, etc. In addition, the use of a single camera does not allow the scale to be known. And finally the detection of keypoints in the image is a time consuming process.

As a consequence, artificial markers have become a popular approach for camera pose estimation, especially the squared planar markers ones [13–17]. They have been developed to be easily used: it is only necessary to print them on a piece of paper and place them in the environment. Their design, based on a black square surrounded by a white area, allows it to be efficiently detected in images. In addition, these markers have four corners with which it is possible to estimate the camera pose.

Despite the recent advances in artificial marker, there are limitations that affect the performance of these systems. First, marker detection is a time-consuming task. In general,

Fig. 1.1 A wide range of applications require accurate real-time location, augmented reality and autonomous navigation are some examples.

the detection time is directly proportional to the size of the image analyzed. Considering that pose estimation needs to be performed in systems with low computational resources, such as phones or mobile robots, it is important developing strategies to optimize the use of resources, without limiting the accuracy of pose estimation.

Second, the marker detection process is sensitive to noise, blurring and occlusion. The blurring effect on the image is mainly produced by camera movement. Even when moving the camera at low speed, the presence of blurring appears limiting the detection capabilities (see Fig.1.2). On the other hand, environment objects may appear in the scene occluding the marker. Finally, the detection of a marker is limited to a range of distances that depends on the size of the marker itself. It means that large markers can be detected from farther distances, but as the camera approaches, the marker is no longer completely visible and therefore can not be detected (*resolution problem*) (see Fig.1.3).

This thesis deals with the aforementioned problems, according to the following schedule. The first contribution (see Chapter 2) proposes a novel strategy to improve the efficiency of marker-based systems in terms of computation time. Employing a multi-scale representation of the image, our algorithm automatically adapts and determines the optimal scale level for both detection, marker identification, as well as an upsampling strategy for subpixel accurate localization of marker corners. As a second contribution (see Chapter 3), we propose a

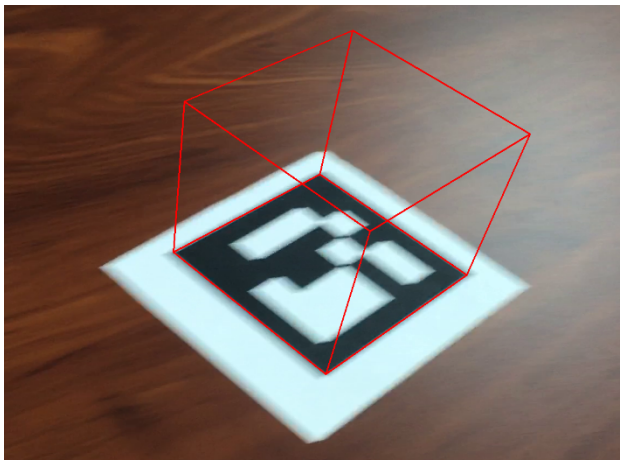


Fig. 1.2 ArUco marker detection under challenging conditions. We propose a novel method for marker detection under blurring conditions produced by camera motion.

new marker design as well as a keypoint-based detection methodology, which solves severe occlusion problems while at the same time achieving a wider range of detection distances than traditional marker systems. And as a third and final contribution (see Chapter 4), we propose a novel strategy for artificial marker detection based on Discriminative Correlation Filters (DCF), improving detection capabilities even in the presence of extreme blurring conditions.

The remaining sections of this introductory chapter are as follows. Section 1.1 shows a summary of the *camera pose estimation* problem. Section 1.2 analyzes the use of fiducial markers to sort it out. And finally, Section 1.3 shows the objectives proposed in the development of the thesis and the proposed contributions.

1.1 Camera pose estimation

Camera pose estimation is the problem of determining the location and orientation of a camera with respect to a reference coordinate system. It can be tackled by assuming the principle of perspective-camera model, or the pinhole model [18]. Generally speaking, the equation that relates a 3D point referenced in a global reference system and its projection on the image plane can be expressed as:

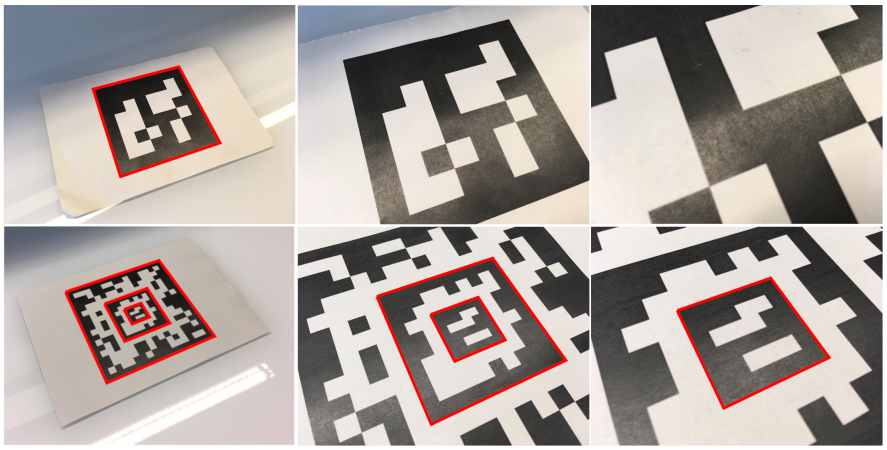


Fig. 1.3 Top row shows the resolution problem. Squared markers are detected under a limited resolution range. Bottom row shows our proposal, the Fractal Marker, that is more robust to occlusion and is detectable under a wider range of resolutions.

$$h \cdot \begin{bmatrix} p \\ 1 \end{bmatrix} = K \cdot E \cdot \begin{bmatrix} P \\ 1 \end{bmatrix} \tag{1.1}$$

where, $p = (x, y)$ is the projection on the image plane (pixel coordinates) of the three-dimentional point $P \in \mathbb{R}^3$, h is a scaling factor and K refers to the intrinsic camera parameters, namely the focal lengths (f_x, f_y) and optical centers (c_x, c_y), that are obtained using a calibration process [19].

$$K = \begin{bmatrix} \mathrm{f}_x & 0 & \mathrm{c}_x & 0 \\ 0 & \mathrm{f}_y & \mathrm{c}_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1.2}$$

Finally, the 4×4 matrix E denotes the extrinsic parameters. It is constituted by the union the 3×3 rotation matrix R, as well as the translation 3D vector t. In other words, E is the Euclidean transformation of the point $P \in \mathbb{R}^3$ from the world reference system to the camera reference system

$$E = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}. \tag{1.3}$$

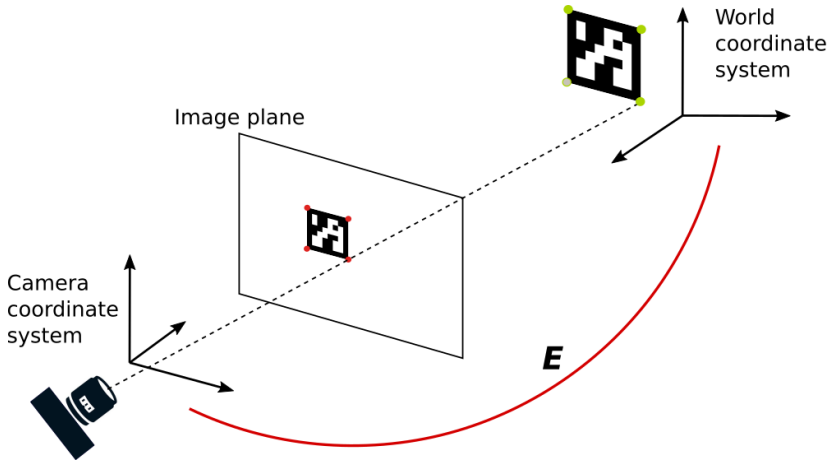


Fig. 1.4 Camera pose estimation from the four detected points of a marker. The transformation between the two coordinate systems, is based on finding the E that minimizes the reprojection error of a set of 3D points on the image plane.

Since real cameras suffers from lenses distortion [19], the ideal pin-hole projection differs from the real projected location. To account for that discrepancy, the distortion θ parameters must be applied. Thus, we define the real projection of a point P by

$$p = \psi(\theta, K, E, P), \tag{1.4}$$

which accounts for that distortion.

Estimating the camera pose (the extrinsic matrix E) can be dealt as the Perspective-n-Points (PnP) problem. Given a set of known 3D points and their respective 2D projections, it is possible to find the transform E that relate them, see Fig. 1.4.

This problem can be bounded to a process of minimization of the reprojection error of the set of points observed in the image:

$$E = \arg\min_{\hat{E}} \sum_{i=1}^{n} (\psi(\theta, K, \hat{E}, P_i) - p_i)^2 \tag{1.5}$$

Thus, a non-linear iterative optimization using Levenberg-Marquardt algorithm [20, 21] can be employed to obtain the camera pose E.

A critical aspect in camera pose estimation is to determine which scene points to be used for optimization. There are two methodologies to establish correspondences between recognizable points in the environment and their respective projections in the image: using natural landmarks or artificial markers.

- *Natural landmarks* are distinctive regions of interest of the environment that are easily recognizable in the images. The term keypoint is normally employed to describe these areas in the image, and special descriptors are employed to encode its visual properties. Thus, descriptors are used to establish correspondences between the 3D real world position and their corresponding 2D projections.

  Although many studies have been carried out in recent years on the use of natural landmarks [10, 11, 22], they have limitations in areas with low texture and demand a high computational power for their computation.

- *Artificial markers*, on the other hand, are designed to be easily detected in images. Among the most commonly used artificial markers are the square planar ones. Generally, these are printed on a piece of paper and incorporated into the scene (see Fig. 1.1). In this case, reference points of the marker, such as its corners, are used to do camera pose estimation [23, 15, 13].

The development of this thesis is focused on camera pose estimation, mainly by using artificial markers.

1.2 Artificial markers

Artificial markers, also known as fiducial markers, are widely used to estimate the camera pose due to their robustness, accuracy and speed [15, 13, 16, 24]. An artificial marker system is composed of a predefined set of artificial markers, as well as the methods and tools that facilitates their detection. In the simplest case, markers can be defined as planar dots, LEDs or reflective spheres [25, 26]. However, they require a complex identification system based on the relative position between markers. An evolution of the previous markers are the circular

planar markers, which incorporate the identification in sectors or concentric rings [27, 28], and also 2D barcodes [29]. However, among the most popular markers are square planar markers [15, 13, 24]. These are composed of a black outer border, surrounded by a white border, and coded information inside which allows each marker to be uniquely identified. The main advantage of these markers is that the camera pose can be estimated using the four corners of a single marker.

The marker detection process is carried out by first thresholding the scene, from which a set of quadrilateral polygon regions (candidate markers) are extracted from the background. Then, the internal region of each of the polygons is analyzed to validate or discard the marker, and finally, the four corners of the polygon are used to estimate the camera pose using the Levenberg-Marquardt optimization process [20, 21].

Among the steps involved, image thresholding and marker identification are the most time-consuming operations. The thresholding time depends on the image resolution, thus, larger images requires more time to complete the task. On the other hand, in order to identify the markers, the system must first create a canonical image of each contour detected in the image, where the time spent on each region will depend on the size of the contour in the original image. In Chapter 2, we show a novel approach to minimize computational time by addressing these problems.

Despite the advances in marker-based systems, there are still some limitations in terms of efficiency and robustness. The occlusion problem is tackled by ArUco library where it is partially solved using a board composed of multiple markers [13]. Also, the system Apriltag3 [30] proposes the use of configurable markers which can be adapted to deal with problems such as occlusion. Furthermore, HArCo [31] presents a new hierarchical marker structure, where in some white marker cells are used to embed new layer of markers. In Chapter 3, we propose the use of a new marker structure, the Fractal Marker, composed of multiple markers which are recursively nested. Unlike a marker board where each marker has a distance from the center of the board, in this case the markers share the same center. With this new approach, we solve the occlusion problem, in addition to solving the resolution problem, allowing a marker to be detected from a wide range of distances, see Fig. 1.3. Besides, the

detection process contemplates the case where no marker has been detected, as occurs in severe occlusion. In this case, the system adopts a keypoint-based approach considering the internal corners of the marker.

Finally, most artificial markers methods have trouble dealing with blurring. However, the presence of blurring in the image is unavoidable; unstabilized camera movements lead to problems of marker detection and identification. For this purpose, Chapter 4 proposes a novel methodology for detection and tracking of fiducial markers based on *Discriminative Correlation Filters (DCF)*. In this case, the image areas corresponding to the marker and its respective corners are used to initialize correlation filters, which represented in the frequency domain allow to speed up and provide robustness to the marker detection process (see Fig. 1.2).

1.3 Objectives and contributions

The objectives of this thesis are aimed at improving the detection capabilities of marker-based systems, allowing accurate camera pose estimations. In this sense, a set of methodologies and strategies have been proposed and developed through the following works:

- **Speeded up detection of squared fiducial markers [32]**

  A new strategy to speed up the marker detection process while preserving accuracy and robustness is presented. From a multi-scale representation of the image, the system strategically determines which resolution to work with. Therefore, the proposed system use smaller resolutions of the image to perform time-consuming processes such as marker detection, and using higher resolutions to validate the accuracy of their corners needed for estimation.

  Let me point out that it is the most cited article in Image and Vision Computing journal since 2018, receiving the Editors Choice Award 2020.

- **Fractal Markers: a new approach for long-range camera pose estimation under occlusion [33]**

A new artificial marker design is presented, the Fractal Marker, in which several markers are nested recursively. The configuration of the marker allows it to be detected over a wider range of distances than traditional markers. In addition, we present a keypoint-based strategy using all corners of the marker, which facilitates marker detection even under severe occlusion conditions.

- **Tracking fiducial markers with Discriminative Correlation Filters [34]**

  A novel marker detection and tracking technique using Discriminative Correlation Filters is proposed. Markers and their corners are represented in the frequency domain using the Fourier transform. With this new approach, we increase the detection capability of marker-based systems in real environments. In this case, markers can be observed even in adverse blurring conditions. In addition, we propose a new localization strategy based on a marker map. It compares with the main state-of-art localization and mapping methods, providing better results in terms of accuracy, while maintaining computational speeds.

Chapter 2

First contribution. "Speeded up detection of squared fiducial markers"

Image and Vision Computing 76 (2018) 38–47

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Speeded up detection of squared fiducial markers ☆

Francisco J. Romero-Ramirez[a], Rafael Muñoz-Salinas[a,b,*], Rafael Medina-Carnicer[a,b]

[a] Departamento de Informática y Análisis Numérico, Edificio Einstein, Campus de Rabanales, Universidad de Cofdoba, 14071 Córdoba, Spain
[b] Instituto Maimónides de Investigación en Biomedicina (IMIBIC), Avenida Menéndez Pidal s/n, 14004 Córdoba, Spain

ARTICLE INFO

Article history:
Received 27 November 2017
Received in revised form 22 March 2018
Accepted 26 May 2018
Available online 15 June 2018

Keywords:
Fiducial markers
Marker mapping
SLAM

ABSTRACT

Squared planar markers have become a popular method for pose estimation in applications such as autonomous robots, unmanned vehicles and virtual trainers. The markers allow estimating the position of a monocular camera with minimal cost, high robustness, and speed. One only needs to create markers with a regular printer, place them in the desired environment so as to cover the working area, and then registering their location from a set of images.

Nevertheless, marker detection is a time-consuming process, especially as the image dimensions grows. Modern cameras are able to acquire high resolutions images, but fiducial marker systems are not adapted in terms of computing speed. This paper proposes a multi-scale strategy for speeding up marker detection in video sequences by wisely selecting the most appropriate scale for detection, identification and corner estimation. The experiments conducted show that the proposed approach outperforms the state-of-the-art methods without sacrificing accuracy or robustness. Our method is up to 40 times faster than the state-of-the-art method, achieving over 1000 fps in 4 K images without any parallelization.

1. Introduction

Pose estimation is a common task for many applications such as autonomous robots [1-3], unmanned vehicles [4-8] and virtual assistants [9-12], among others.

Cameras are cheap sensors that can be effectively used for this task. In the ideal case, natural features such as keypoints and texture [13-16] are be employed to create a map of the environment. Although some of the traditional problems of previous methods for this task have been solved in the last few years, other problems remain. For instance, they are subject to filter stability issues or significant computational requirements.

In any case, artificial landmarks are a popular approach for camera pose estimation. Square fiducial markers, comprised by an external squared black border and an internal identification code, are especially attractive because the camera pose can be estimated from the four corners of a single marker [17-20]. The recent work of Muñoz-Salinas et al. [21] is a step forward in the use of this type of markers in large-scale problems. One only needs to print the set of markers with a regular printer, place them in the area under which the camera must move, and take a set of pictures of the markers. The pictures are then analyzed and the three-dimensional marker locations automatically obtained. Afterward, a single image spotting a marker is enough to estimate the camera pose.

Despite the recent advances, marker detection can be a time-consuming process. Considering that the systems requiring localization have in many cases limited resources, such as mobile phones and aerial vehicles, the computational effort of localization should be kept to a minimum. The computing time employed in marker detection is a function of the image size employed: the larger the images, the slower the process. On the other hand, high-resolution images are preferable since markers can be detected, even if they are far from the camera, with high accuracy. The continuous reduction in the cost of the cameras, along with the increase of their resolution, makes it necessary to develop methods able to reliably detect the markers in high-resolution images.

The main contribution of this paper is a novel method for detecting square fiducial markers in video sequences. The proposed method relies on the idea that markers can be detected in smaller versions of the image, and employs a multi-scale approach to speed up computation while maintaining the precision and accuracy. In addition, the system is able to dynamically adapt its parameters in order to achieve maximum performance in the analyzed video sequence. Our approach has been extensively tested and compared with the state-of-the-art methods for marker detection. The results show that our method is more than an order of magnitude faster than state-of-the-art approaches without compromising robustness or accuracy, and without requiring any type of parallelism.

☆ This paper has been recommended for acceptance by Luis Merino.

* Corresponding author.
E-mail addresses: fj.romero@uco.es (F.J. Romero-Ramirez), in1musar@uco.es (R. Muñoz-Salinas), rmedina@uco.es (R. Medina-Carnicer).

https://doi.org/10.1016/j.imavis.2018.05.004
0262-8856/

The remainder of this paper is structured as follows. Section 2 explains the works most related to ours. Section 3 details our proposal for speeding up the detection of markers. Finally, Section 4 gives a exhaustive analysis of the proposed method and Section 5 draws some conclusions.

2. Related works

Fiducials marker systems are commonly used for camera localization and tracking when robustness, precision, and speed are required. In the simplest case, points are used as fiducial markers, such as LEDs, retroreflective spheres and planar dots [22,23]. However, their main drawback is the need of a method to solve the assignment problem, i.e., assigning a unique and consistent identifier to each element over time. In order to ease the problem, a common solution consists in adding an identifying code into each marker. Examples of this are planar circular markers [24,25], 2D-barcodes [26,27] and even some authors have proposed markers designed using evolutionary algorithms [28].

Among all proposed approaches, those based on squared planar markers have gained popularity. These markers consist of an external black border and an internal code (most often binary) that uniquely identifies each marker (see Fig. 1). Their main advantage is that the pose of the camera can be estimated from a single marker.

ARToolKit [29] is one of the pioneer proposals. They employed markers with a custom pattern that is identified by template matching. This identification method, however, is prone to error and not very robust to illumination changes. In addition, the method's sensitivity degrades as the number of markers increases. As a consequence, other authors improved that work by using binary BCH codes [30] (which allows a more robust error detection) and named it ARToolKit+ [31]. The project was halted and followed by the Studierstube Tracker project [32], which is privative. Similar to the ARToolKit+ project is the discontinued project ARTag [33].

BinARyID [34] is one of the first systems that proposed a method for generating customizable marker codes. Instead of using a predefined set of codes, they proposed a method for generating the desired number of codes for each particular application. However, they do not consider the possibility of error detection and correction. AprilTags [18], however, proposed methods for error detection and correction, but their approach was not suitable for a large number of markers.

The work ArUco [17] is probably the most popular system for marker detection nowadays. It adapts to non-uniform illumination, and is very robust, being able to do error detection and correction of the binary codes implemented. In addition, the authors proposed a method to obtain optimal binary codes (in terms of intermarker-distance) using Mixed Integer Linear Programming [35]. Chilitags [36] is a variation of ArUco that employs a simpler method for decoding the marker binary codes. As we show in the Experiments and results section, the method has a bad behavior in high-resolution images.

The recent work [21] is a step towards the applicability of such methods to large areas, proposing a method for estimating the three-dimensional location of a set of markers freely placed in the environment (Fig. 1). Given a set of images taken with a regular camera (such as a mobile phone), the method automatically estimates their location. This is an important step that allows extending the robust localization of fiducial markers to very large areas.

Although all fiducial marker systems aim maximum speed in their design, few specific solutions have been proposed to speed up the detection process. The work of Johnston et. al. [37] is an interesting example in which the authors propose a method to speed up computation by parallelizing the image segmentation process. Nevertheless, both speed and computing power is a crucial aspect, especially if the localization system needs to be embedded in devices with limited resources.

Our work can be seen as an improvement of the ArUco system, that according to our experience, is one of the most reliable fiducial

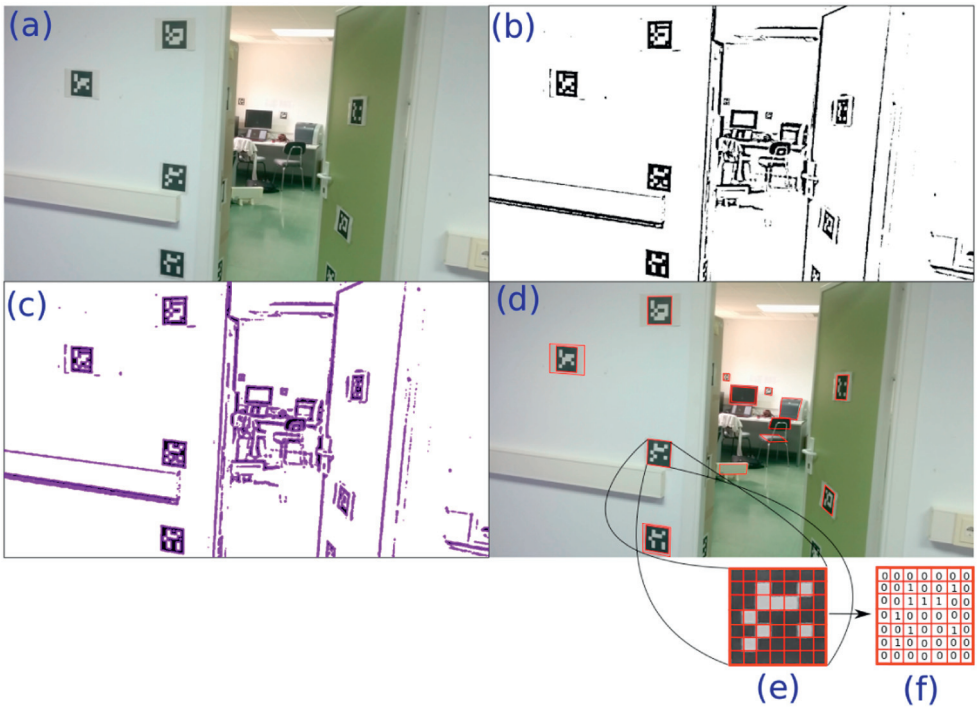


Fig. 1. Detection and identification pipeline of ArUco. (a) Original image. (b) Image thresholded using an adaptive method. (c) Contours extracted. (d) Filtered contours that approximate to four-corner polygons. (e) Canonical image computed for one of the squared contours detected. (f) Binarization after applying Otsu's method.

marker systems nowadays (see Section 4 for further details). We propose a novel method for marker detection and identification that allows to speed up the computing time in video sequences by wisely exploiting temporal information and an applying multi-scale approach. In contrast to previous works, no parallelization is required in our method, thus making it especially attractive for mobile devices with limited computational resources.

3. Speeded up marker detection

This section provides a detailed explanation of the method proposed for speeding up the detection of squared planar markers. First, Section 3.1 provides an overview of the pipeline employed in the previous work, ArUco [17], for marker detection and identification, highlighting the parts of the process susceptible to be accelerated. Then, Section 3.2 explains the proposed method to speed up the process.

3.1. Marker detection and identification in ArUco

The main steps for marker detection and identification proposed in ArUco [17] are depicted in Fig. 1. Given the input image I (Fig. 1a), the following steps are taken:

- Image segmentation (Fig. 1b). Since the designed markers have an external black border surrounded by a white space, the borders can be found by segmentation. In their approach, a local adaptive method is employed: the mean intensity value m of each pixel is computed using a window size w_t. The pixel is set to zero if its intensity is greater than $m - c$, where c is a constant value. This method is robust and obtains good results for a wide range of values of its parameters w_t and c.
- Contour extraction and filtering (Fig. 1 (c, d)). The contour following algorithm of Suzuki and Abe [38] is employed to obtain the set of contours from the thresholded image. Since most of the contours extracted correspond to irrelevant background elements, a filtering step is required. First, contours that are too small are discarded. Second, the remaining contours are approximated to its most similar polygon using the Douglas and Peucker algorithm [39]. Those that do not approximate well to a four-corner convex polygon are discarded from further processing.
- Marker code extraction (Fig. 1 (e, f)). The next step consists in analyzing the inner region of the remaining contours to determine which of them are valid markers. To do so, perspective projection is first removed by computing the homography matrix, and the resulting canonical image (Fig. 1e) is thresholded using the Otsu's method [40]. The binarized image (Fig. 1f) is divided into a regular grid and each element is assigned a binary value according to the majority of the pixels in the cell. For each marker candidate, it is necessary to determine whether it belongs to the set of valid markers or if it is a background element. Four possible identifiers are obtained for each candidate, corresponding to the four possible rotations of the canonical image. If any of the identifiers belong to the set of valid markers, then it is accepted.
- Subpixel corner refinement. The last step consists in estimating the location of the corners with subpixel accuracy. To do so, the method employs a linear regression of the marker's contour pixels. In other words, it estimates the lines of the marker sides employing all the contour pixels and computes the intersections. This method, however, is not reliable for uncalibrated cameras with small focal lenses (such as fisheye cameras) since they usually exhibit high distortion.

When analyzing the computing times of this pipeline, it can be observed that the Image segmentation and the Marker code extraction steps are consuming most of the computing time. The time employed in the image segmentation step is proportional to the image size, that also influences the length of the contours extracted and thus the computing time employed in the Contour extraction and filtering step. The extraction of the canonical image (in the Marker code extraction step) involves two operations. First is the computation of the homography matrix, which is cheap. But then, the inner region of each contour must be warped to create the canonical image. This step requires access to the image pixels of the contour region performing an interpolation in order to obtain the canonical image. The main problem is that the time required to obtain the canonical image depends on the size of the observed contour. The larger a contour in the original image, the more time it is required to obtain the canonical image. Moreover, since most of the contours obtained do not belong to markers, the system may employ a large amount of time computing canonical images that will be later rejected.

A simpler approach to solving that problem would be to directly sample a few sets of pixels from the inner region of the marker. This is the method employed in ChiliTags. However, as it will be shown in the Experiments and results section, it is prone to many false negatives.

3.2. Proposed method

The key ideas of our proposal in order to speed up the computation are explained below. First, while the adaptive thresholding method employed in ArUco is robust to many illumination conditions without altering its parameters, it is a time-consuming process that requires a convolution. By taking advantage of temporal information, the adaptive thresholding method is replaced by a global thresholding approach.

Second, instead of using the original input image, a smaller version is employed. This is based on the fact that, in most cases, the useful markers for camera pose estimation must have a minimum size. Imagine an image of dimensions 1920×1080 pixels, in which a marker is detected as a small square with a side length of 10 pixels. Indeed, the estimation of the camera pose is not reliable at such small resolution. Thus, one might want to set a minimum length to the markers employed for camera pose estimation. For instance, lets say that we only use markers with a minimum side length of $\dot{\tau}_i = 100$ pixels, i.e., with a total area of 10,000 pixels. Another situation in which we can set a limit to the length of markers is when processing video sequences. It is clear that the length of a marker must be similar to its length in the previous frame.

Now, let us also think about the size of the canonical images employed (Fig. 1e). The smaller the image, the faster the detection process but the poorer the image quality. Our experience, however, indicates that very reliable detection of the binary code can be obtained from very small canonical images, such as 32×32 pixels. In other words, all the rectangles detected in the image, no matter their side length, are reduced to canonical images of side length $\tau_c = 32$ pixels, for the purpose of identification.

Our idea, then, is to employ a reduced version of the input image, using the scale factor $\frac{\tau_c}{\dot{\tau}_i}$, so as to speed up the segmentation step. In the reduced image, the smallest allowed markers, with a side length of 100 pixels in the original image, appear as rectangles with a side length of 32 pixels. As a consequence, there will be no loss of quality when they are converted into the canonical image.

This idea has one drawback: the location of the corners extracted in the low resolution image is not as good estimations as the ones that can be obtained in the original image. Thus, the pose estimated with them will have a higher error. To solve that problem, a corner upsampling step is included, in which the precision of the corners

is refined up to subpixel accuracy in the original input image by employing an image pyramid.

Finally, it must be considered that the generation of the canonical image is a very time-consuming operation (even if the process is done in the reduced image) that is proportional to the contour length. We propose a method to perform the extraction of the canonical images in almost constant time (independently of the contour length) by wisely employing the image pyramid.

Below, there is a detailed explanation of the main steps of the proposed method, using Fig. 2 to ease the explanation.

1. *Image resize:* Given the input image I (Fig. 2a), the first step consists in obtaining a resized version I^r (Fig. 2b) that will be employed for segmentation. As previously pointed out, the size of the reduced image is calculated as:

$$I_w^r = \frac{\tau_c}{\dot{\tau}_i} I_w \Big\rfloor ; I_h^r = \frac{\tau_c}{\dot{\tau}_i} I_h \Big\rfloor , \tag{1}$$

where the subscripts $_w$ and $_h$ denote width and height respectively. In order to decouple the desired minimum marker size from the input image dimensions, we define $\dot{\tau}_i$ as:

$$\dot{\tau}_i = \tau_c + \max(I_w, I_h)\tau_i \mid \tau_i \in [0,1], \tag{2}$$

where the normalized parameter τ_i indicates the minimum marker size as a value in the range $[0,1]$. When $\tau_i = 0$, the reduced image will be the same size as the original image. As τ_i tends to one, the image I^r becomes smaller, and consequently, the computational time required for the following step is reduced. The impact of this parameter in the final speed up is measured in the Experiments and results section.

2. *Image segmentation:* As already indicated, a global threshold method is employed using the following policy. If no markers were detected in the previous frame, a random threshold search is performed. The random process is repeated up to three times using the range of threshold values $[10,240]$. For each tested threshold value, the whole pipeline explained below is performed. If after a number of attempts, no marker is found, it is assumed that no markers are visible in the frame. If at least one marker is detected, a histogram is created using the pixel values of all detected markers. Then, Otsu's algorithm [40] is employed to select the optimal threshold for the next frame. The calculated threshold is applied to I^r in order to obtain I^t (Fig. 2c). As we show experimentally, the proposed method can adapt to smooth and abrupt illumination changes.
3. *Contour extraction and filtering:* First, contours are extracted from the image I^t using Suzuki and Abe algorithm [38], then small contours are removed. Since the extracted contours will rarely be squared (due to perspective projection), their perimeter is employed for rejection purposes: those with a perimeter smaller than $P(\tau_c) = 4 \times \tau_c$ pixels are rejected. For the remaining contours, a polygonal approximation is performed using Douglas and Peucker algorithm [39], and those that do not approximate to a convex polygon of four corners are also rejected. Finally, the remaining contours are the candidates to be markers (Fig. 2d).
4. *Image pyramid creation:* An image pyramid

$$\mathcal{I} = (I^0, \ldots, I^n)$$

with a set of resized versions of I, is created. I^0 denotes the original image and the subsequent images I^i are created by subsampling I^{i-1} by a factor of two.
The number n of images in the pyramid is such that the smallest image dimensions is close to $\tau_c \times \tau_c$, i.e.,

$$n = \underset{v \mid I^v \in \mathcal{I}}{\operatorname{argmin}} \left| (I_w^v I_h^v) - \tau_c^2 \right| . \tag{3}$$

5. *Marker code extraction:* In this step, the canonical images of the remaining contours must be extracted and then binarized. Our method uses the pyramid of images $\mathcal{I}$ previously computed to ensure that the process is performed in constant time, independently of the input image and contour sizes. The key principle is selecting, for each contour, the image from the pyramid in which the contour length is most similar to the canonical image length $P(\tau_c)$. In this manner, warping is faster.
Let us consider a detected contour $\vartheta \in I^r$, and denote by $P(\vartheta)^j$ its perimeter in the image $I^j \in \mathcal{I}$. Then, the best image $I^h \in \mathcal{I}$ for homography computation is selected as:

$$I^h \mid h = \underset{j \in \{0,1,\ldots n\}}{\operatorname{argmin}} \left| P(\vartheta)^j - P(\tau_c) \right| . \tag{4}$$

The pyramidal warping method employed can be better understood in Fig. 3, which shows a scene with three markers at different distances. The left images represent the canonical images obtained while the right images show the pyramid of images. In our method, the canonical image of the smallest marker is extracted from the largest image in the pyramid (top row of Fig. 3). As the length of the marker increases, smaller images of the pyramid are employed to obtain the

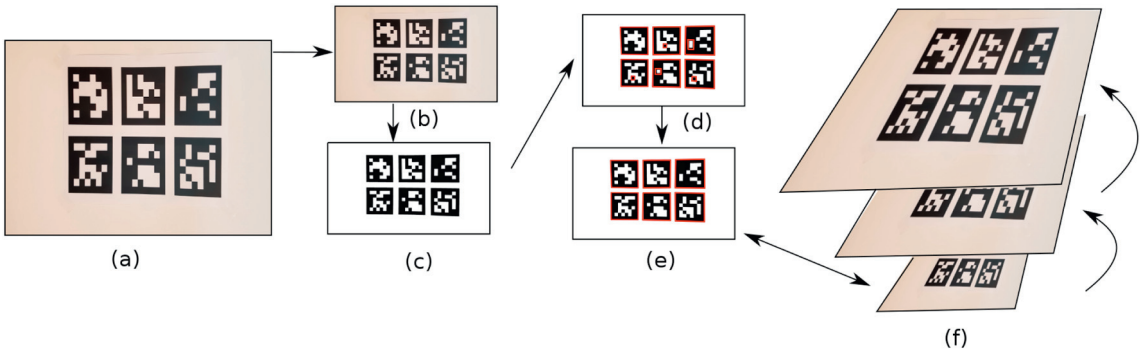


Fig. 2. Process pipeline. Main steps for fast detection and identification of squared planar markers. (a) Original input image. (b) Resized image for marker search. (c) Thresholded image. (d) Rectangles found (pink). (e) Markers detected with its corresponding identification. The image pyramid is used to speed up homography computation. (f) The corners obtained in (e) are upsampled to find their location in the original image with subpixel precision.

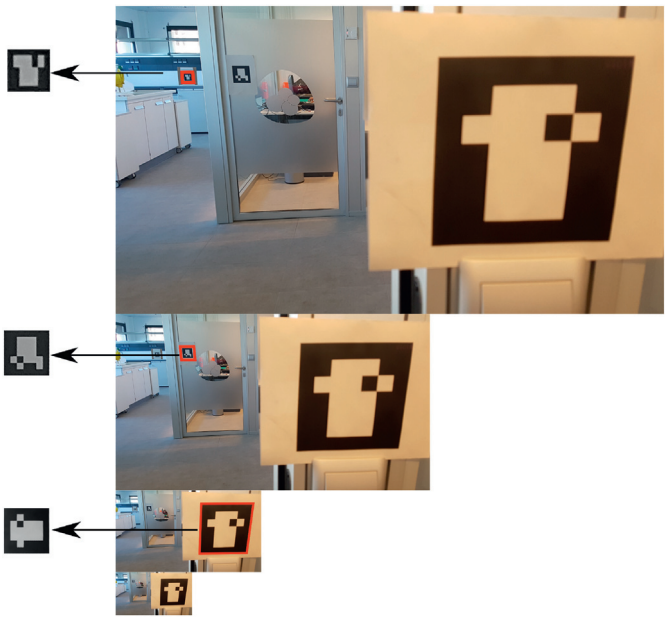


Fig. 3. Pyramidal warping. Scene showing tree marker at different resolutions. The left column shows the canonical images warped from the pyramid of images. Larger markers are warped from smaller images. For each marker, the image of the pyramid that minimizes the warping time while preserving the resolution is selected.

canonical view. This guarantees that the canonical image is obtained in almost constant time using the minimum possible computation.

Finally, for each canonical image, Otsu's method [40] for binarization is employed, and the inner code analyzed to determine whether it is a valid marker or not. This is a very cheap operation.

6. *Corner upsampling:* So far, markers have been detected in the image I^r. However, it is required to precisely localize their corners in the original image I. As previously indicated, the precision of the estimated camera pose is directly influenced by the precision in the corner localization. Since the difference in size between the images I and I^r can be very large, a direct upsampling can lead to errors. Instead, we proceed in incremental steps looking for the corners in larger versions of the image I^r until the image I is reached.

   For the corner upsampling task, the image $I^i \in \mathcal{I}$ of the pyramid with the most similar size to I^r is selected in the first place, i.e.,

$$I^i = \underset{I^v \in \mathcal{I}}{\operatorname{argmin}} \left| (I^v_w I^v_h) - (I^r_w I^r_h) \right|. \tag{5}$$

   Then, the position of each contour corner in the image I^i is computed by simply upsampling the corner locations. This is, however, an approximate estimation that does not precisely indicate the corner position in the image I^i. Thus, a corner refinement process is done in the vicinity of each corner so as to find its best location in the selected image I^i. For that purpose, the method implemented in the OpenCV library [41] has been employed. Once the search is done in I^i for all corners, the operation is repeated for the image I^{i-1}, until I^0 is reached. In contrast to the ArUco approach, this one is not affected by lens distortions.

7. *Estimation of τ_i:* The parameter τ_i has a direct influence in the computation time. The higher it is, the faster the computation. A naive approach consists in setting a fixed value for this parameter. However, when processing video sequences, the parameter can be automatically adjusted at the end of each frame. In the first image of the sequence, the parameter τ_i is set to zero. Thus, markers of any size are detected. Then, for the next frame, τ_i is set to a value slightly smaller than the size of the smallest marker detected in the previous frame. In this way, markers could be detected even if the camera moves away from them. Therefore, the parameter τ_i can be dynamically updated as:

$$\tau_i = (1 - \tau_s) P(\vartheta^s)/4 \tag{6}$$

   where ϑ^s is the marker with the smallest perimeter found in the image, and τ_s is a factor in the range $(0, 1]$ that accounts for the camera motion speed. For instance, when $\tau_s = 0.1$, it means that in the next frame, τ_i is such that markers 10% smaller than the smallest marker in the current image will be sought. If no markers are detected in a frame, τ_i is set to zero so that in the next frame markers of any size can be detected.

As can be observed, the proposed pipeline includes a number of differences with respect to the original ArUco pipeline that allows increasing significantly the processing speed as we show next.

4. Experiments and results

This section shows the results obtained to validate the methodology proposed for the detection of fiducial markers.

First, in Section 4.1, the computing times of our proposal are compared to the best alternatives found in the literature: AprilTags [18], ChiliTags [36], ArToolKit+ [31], as well as ArUco [17] which is included in the OpenCV library[1]. Then, Section 4.2 analyzes and compares the sensitivity of the proposed method with the above-mentioned methods. The main goal is to demonstrate that our approach is able to reliably detect the markers with a very high true positive ratio, under a wide range of marker resolutions, while keeping the false positive rate to zero. Afterward, Section 4.3 studies the impact of the different system parameters on the speed and sensitivity, while Section 4.4 evaluates the precision in the estimation of the corners. Finally, Section 4.5 shows the performance of the proposed method in a realistic video sequence with occlusions, illumination, and scale changes.

To carry out the first three experiments, several videos have been recorded in our laboratory. Fig. 4(b–e) shows some images of the video sequences employed. For these tests, a panel with a total of 16 markers was printed (Fig. 4a), four from each one of the fiducial markers employed. The sequences were recorded at different distances at a frame rate of 30 fps using an Honor 5 mobile phone at 4 K resolution. The videos employed are publicly available[2] for evaluation purposes.

In the video, there are frames in which the markers appear as small as can be observed in Fig. 4b, where the area of each marker occupies only 0.5% of the image, and frames in which the marker is observed as big as in Fig. 4e, where the marker occupies 40% of total image area. In total, the video sequences recorded sum up to 10,666 frames. The video frames have been processed at different resolutions so that the impact of the image resolution in the computing time can be analyzed. In particular, the following standard image resolutions have been employed: 2016 p (3840 × 2160),

[1] https://opencv.org/.
[2] https://mega.nz/#F!DnA1wIAQ!6f6owb81G0E7Sw3EfddUXQ.

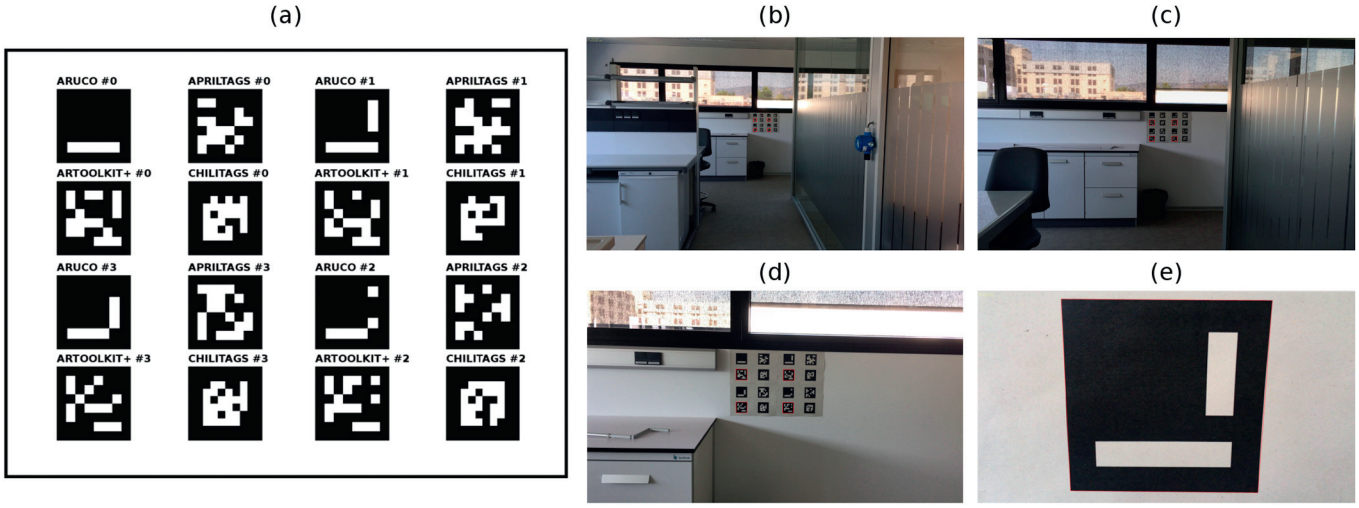


Fig. 4. Test sequences. (a) The set of 16 markers employed for evaluation. There are four markers from each method tested: ArUco, AprilTags, ArToolKit+ and ChiliTags. (b–e) Images from the video sequences used for testing. The markers are seen as small as in (b), and as big as in (e), where the marker represents the 40% of the total image area.

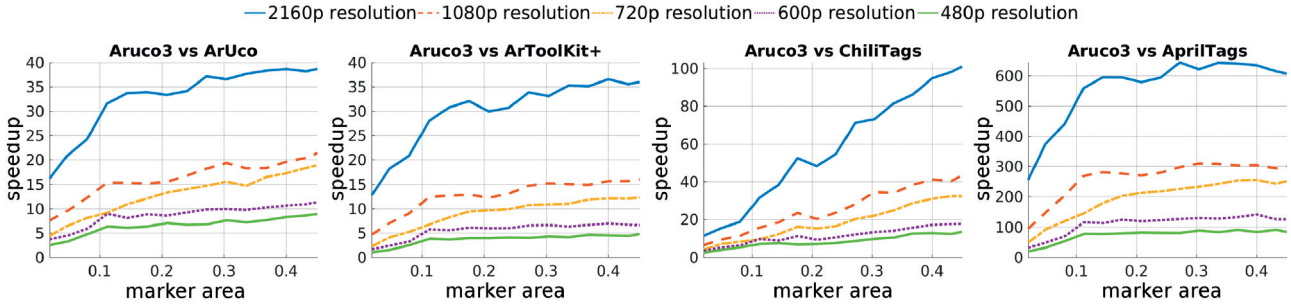


Fig. 5. Speedup of ArUco3 compared to ArUco, ArToolKit+, ChiliTags and AprilTags for resolutions: 2016 p (3840 × 2160), 1080 p (1920 × 1080), 720 p (1280 × 720), 600 p (800 × 600) and 480 p (640 × 480). The horizontal axis represents the percentage of area occupied by the markers in each frame, and the vertical axis one indicates how many times ArUco3 is faster.

1080 p (1920 × 1080), 720 p (1280 × 720), 600 p (800 × 600) and 480 p (640 × 480).

All tests were performed using an Intel® Core™ i7-4700HQ 8-core processor with 8 GB RAM and Ubuntu 16.04 as the operating system. However, only one execution thread was employed in the tests performed.

It must be indicated that the code generated as part of this work has been publicly released as the version 3 of the popular ArUco library[3]. So, in the Experiments and results section, the method proposed in this paper will be referred to as ArUco3.

4.1. Speedup

This section compares the computing times of the proposed method with the most commonly used alternatives AprilTags, ArToolKit+, ChiliTags, and ArUco. To do so, we compute the speedup of our approach as the ratio between the computing time of an alternative (t_1) and the computing time of ArUco3 (t_2) in processing the same image:

$$SpeedUp = t_1/t_2. \tag{7}$$

In our method, the value $\tau_c = 32$ was employed in all the sequences, while τ_i and the segmentation threshold where automatically computed as explained in the Steps 2 and 7 of the proposed method (Section 3.2).

Fig. 5 shows the speedup of our approach for different image resolutions. The horizontal axis represents the relative area occupied by the marker in the image, while the vertical axis represents the speedup. A total of 30 speed measurements were performed for each image, taking the median computing time for our evaluation. In the tests, the speedup is evaluated as a function of the observed marker area in order to better understand the behavior of our approach.

The tests conducted clearly show that the proposed method (ArUco3) is faster than the rest of the methods and that the speedup increases with the image resolution and with the observed marker area. Compared to ArUco implementation in the OpenCV library, the proposed method is significantly faster, achieving a minimum speedup of 17 in 2016 p resolutions, up to 40 in the best case.

In order to properly analyze the computing times of the different steps of the proposed method (Section 3.2), Table 1 shows a summary for different image resolutions. Likewise, Fig. 6 shows the percentage of the total time required by each step. Please notice that Step 7 (Eq. (6)) has been omitted because its computing time is negligible.

As can be seen, the two most time-consuming operations are Steps 3 and 5. In particular, Step 5 requires special attention, since it proves the validity of the multi-scale method proposed for marker

[3] http://www.uco.es/grupos/ava/node/25.

Table 1
Mean computing times (milliseconds) of the different steps of the proposed method for different resolutions.

	Resolution				
	480 p	600 p	720 p	1080 p	2160 p
Step 1: Image resize	0.037	0.050	0.057	0.068	0.101
Step 2: Image segmentation	0.044	0.048	0.059	0.084	0.351
Step 3: Contour extraction and filtering	0.219	0.250	0.301	0.403	1.109
Step 4: Image pyramid creation	0.037	0.076	0.096	0.186	0.476
Step 5: Marker code extraction	0.510	0.519	0.542	0.547	0.583
Step 6: Corner upsampling	0.058	0.065	0.079	0.096	0.134
Time (ms)	0.903	1.009	1.133	1.384	2.755

warping. It can be observed in the table, that the amount of time employed by Step 5 is constant across all resolutions. In other words, the computing time does not increase significantly with the image resolution. Also notice how the time of Step 3 increases in 2160 p. It is because this step involves operations that depend on the image dimensions, which grow quadratically. An interesting future work is to develop methods reducing the time for contour extraction and filtering in high-resolution images.

In any case, considering the average total computing time, the proposed method achieves in average more than 360 fps in 2016 p resolutions and more than 1000 fps in the lowest resolution, without any parallelism.

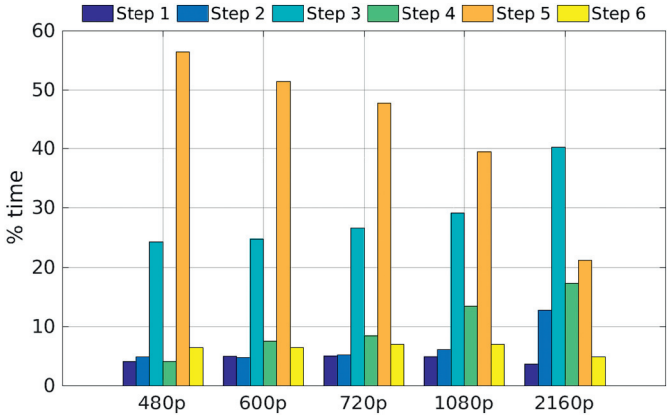


Fig. 6. Main steps ArUco3 times. Percentage of time of the global computation required by each of the steps for resolutions: 2016 p, 1080 p, 720 p, 600 p and 480 p.

4.2. Sensitivity analysis

Correct detection of markers is a critical aspect that must be analyzed to verify that the proposed algorithm is able to obviate redundant information present in the scene, extracting exclusively marker information. Fig. 7 shows the True Positive Rate (TPR) of the proposed method as a function of the area occupied by the marker in the image for different image resolutions.

As can be observed, below certain marker area, the detection is not reliable. This is because the observed marker area is very small, making it difficult to distinguish the different bits of the inner binary code. Once the observed area of the marker reaches a certain limit, the proposed method achieves perfect detection in all resolutions. It must be remarked, that the False Positive Rate is zero in all cases tested. Since it is a binary problem, the True Negative Rate is one (TNR $=$ 1 $-$ FPR).

For a comparative evaluation performance between ArUco3 and the other methods, the TPR has been analyzed individually and the

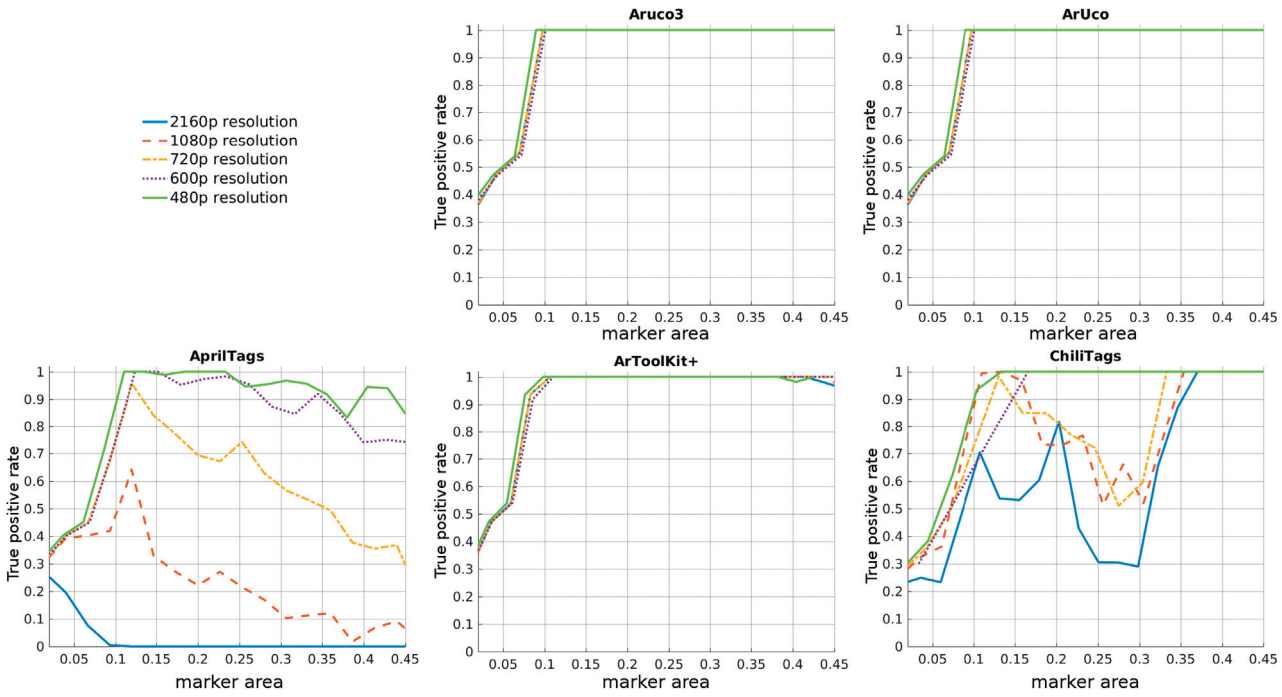


Fig. 7. True positive ratio. Mean true positive ratio (TPR) for ArUco3, Chilitags, ArUco, ArToolKit+ and AprilTags for resolutions: 2016 p, 1080 p, 720 p, 600 p and 480 p), as function of the observed area for the set of markers.

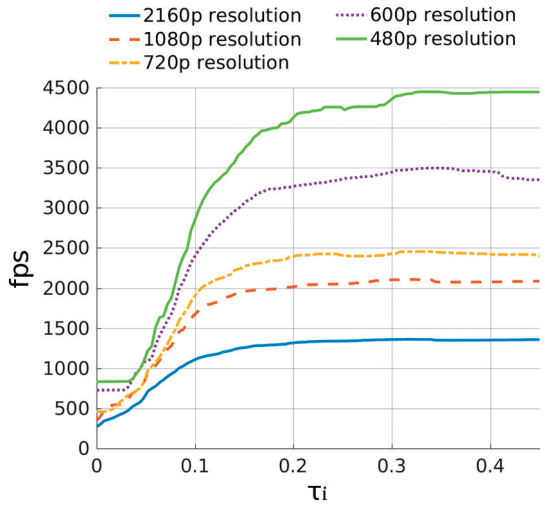


Fig. 8. Parameter τ_i. Speed of method as a function of the parameter τ_i for the different resolutions tested.

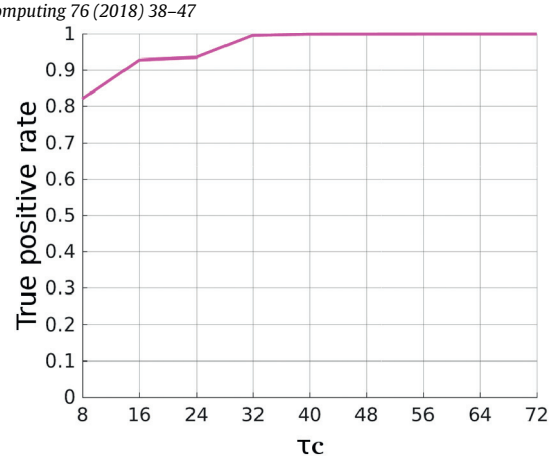


Fig. 9. Parameter τ_c. True positive rate obtained by different configurations of parameter τ_c.

results are shown in Fig. 7. As can be observed, ArUco behaves exactly like ArUco3. AprilTags, however, has very poor behavior in all resolutions, especially as the marker or the image sizes increases. As we already commented in Section 2, AprilTags does not rely on warping the marker image but instead does a subsampling of a few pixels on the image in order to obtain the binary code. This may be one of the reasons for its poor performance. ArToolKit+ behaves reasonably well across all the image resolutions and marker areas, while Chilitags shows a somewhat unreliable behavior in all resolutions but 480 p.

In conclusion, the proposed approach behaves similar to the previous version of ArUco.

4.3. Analysis of parameters

The computing time and robustness of the proposed method depend mainly on two parameters, namely τ_i which indicates the minimum size of the markers detected, and τ_c, the size of the canonical image.

The parameter τ_i has an influence on the computing time, since it determines the size of the resized image I^r (Eq. (1)). We have analyzed the speed as a function of this parameter and the results are shown in Fig. 8. The figure represents the horizontal axis the value τ_i, and in the vertical axis, the average speed (measured as frames per second) in the sequences analyzed, independently of the observed marker area. A different line has been depicted for each image resolution. In this case, we have set fixed the parameter $\tau_c = 32$.

It can be observed that the curves follow a similar pattern in the five cases analyzed. In general, the maximum increase in speed is obtained in the range of values $\tau_i = (0, 0.2)$. Beyond that point, the improvement becomes marginal. To better understand the impact of this parameter, Table 2 shows the reduction of the input image size I for different values of τ_i. For instance, when $\tau_i = 0.02$, the resized image I^r is 48% smaller than the original input image I (see Eq. (1)). Beyond $\tau_i = 0.2$, the resized image is so small that it has no big impact in the speedup because there are other steps with a fixed computing time such as the Step 5 (Marker code extraction).

In any case, it must be noticed that the proposed method is able to achieve 1000 fps in 2016 p resolutions when detecting markers larger than 10% ($\tau_i = 0.1$) of the image area, and the same limit of 1000 fps is achieved for 1080 p resolutions for $\tau_i = 0.05$.

With regard to the parameter τ_c, it indirectly influences the speed since it determines the size of the resized images (Eq. (1)). The smaller it is, the smaller the resized image I^r. Nevertheless, this parameter also has an influence on the correct detection of the markers. The parameter indicates the size of the canonical images used to identify the binary code of markers. If the canonical image is very small, pixels are mixed up, and identification is not robust. Consequently, the goal is to determine the minimum value of τ_c that achieves the best TPR. Fig. 9 shows the TPR obtained for different configurations of the parameter τ_c. As can be seen, for low values of the parameter τ_c (between 8 and 32) the system shows problems in the detection of markers. However, for $\tau_c \geq 32$ there is no improvement in the TPR. Thus, we conclude that the value $\tau_c = 32$ is the best choice.

Table 2
Image size reduction for different values of τ_i.

τ_i	0.01	0.015	0.02	0.1	0.2
Size reduction	0%	31%	48%	82%	90%

4.4. Precision of corner detection

An important aspect to consider in the detection of the markers is *vertex jitter*, which refers to the noise in the estimation of the corners' location. These errors are problematic because they propagate to the estimation of the camera pose. In our method, a corner upsampling step (Step 6 in Section 3.2) is proposed to refine the corners' estimations from the reduced image I^r to the original image I. This section analyzes the proposed method comparing the results with the other marker systems.

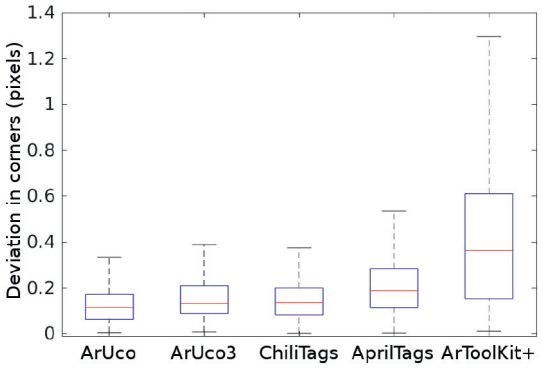


Fig. 10. Vertex jitter measured for the different marker systems.

Table 3
Vertex jitter analysis: Standard deviations of the different methods in estimating the marker corners.

Method	ArUco	ArUco3	Chilitags	AprilTags	ArToolKit+
Average error (pix)	0.140	0.161	0.174	0.225	0.432

In order to perform the experiments, the camera has been placed at a fixed position recording the set of markers already presented in Fig. 4a. Since the camera is not moving, the average location estimated for each corner can be considered to be the correct one (i.e., a Gaussian error distribution is assumed). Then, the standard deviation is an error measure for the localization of the corners. The process has been repeated a total of six times at varying distances and the results obtained are shown in Fig. 10 as box plots. In Table 3, the average error of each method has been indicated.

As can be observed, the ArUco system obtains the best results, followed by our proposal ArUco3. However, it can be seen that the difference between both methods is only 0.02 pixels, which is very small to consider it relevant. Chilitags shows a similar behavior to ArUco and ArUco3, but AprilTags and ArToolKit+ exhibit worse performance.

4.5. Video sequence analysis

This section aims at showing the behavior of the proposed system in a realistic scenario. For that purpose, four markers have been placed in an environment with irregular lighting and a video sequence has been recorded using a 4 K mobile phone camera. Fig. 11(a–e) shows the frames 1, 665, 1300, 1700 and 2100 of the video sequence. At the start of the sequence, the camera is around 5 m away from the markers. The camera approaches the markers and then moves away again. As can be seen, around frame 650 (Fig. 11b), the user occludes the markers temporarily.

Fig. 11f shows the values of the parameter τ_i automatically calculated along the sequence and Fig. 11g the processing speed. As can be observed, the system is able to automatically adapt the value of τ_i according to the observed marker area, thus adapting the computing speed of the system. The maximum speed is obtained around the frame 1300 when the camera is closest to the markers.

It can also be observed that around frame 650 when the user occludes the markers with his hand, the system is unable to detect any marker. Thus, the system searches for the full resolution image ($\tau_i = 0$) and the speed decreases. However, when the markers are observed again, the system recovers its speed.

Finally, Fig. 11h shows the threshold values employed for segmentation in each frame. As can be seen, the system adapts to the illumination changes. Along the sequence, the system does not produce any false negative nor positives.

5. Conclusions and future work

This paper has proposed a novel approach for detecting fiducial markers aimed at maximizing speed while preserving accuracy and robustness. The proposed method is specially designed to take advantage of the increasing camera resolutions available nowadays. Instead of detecting markers in the original image, a smaller version of the image is employed, in which the detection can be done at higher speed. By wisely employing a multi-scale image representation, the proposed method is able to find the position of the marker corners with subpixel accuracy in the original image.

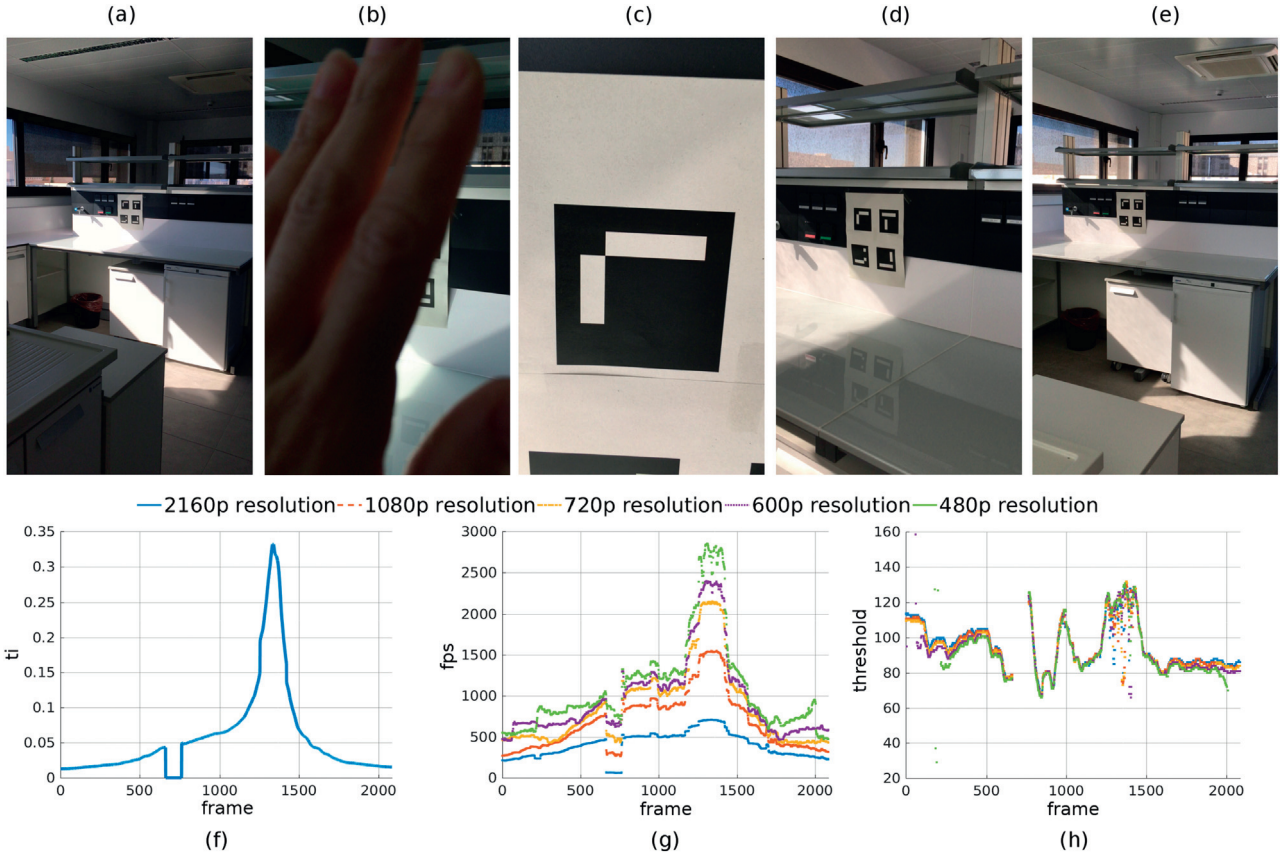


Fig. 11. Video sequence in a realistic scenario. (a–e) Frames of the video sequence. The camera approaches the marker and then moves away. The user occludes the camera temporarily. (f) Evolution of the parameter τ_i automatically computed. (g) Speed of the proposed method in each frame of the sequence. (h) Thresholds automatically computed for each frame. The system adapts to illumination changes.

The size of the processed image, as well as the threshold employed for segmentation, are dynamically adapted in each frame considering the information of the previous one. As a consequence, the system speed dynamically adapts in order to achieve the maximum performance.

As shown experimentally, the proposed method outperforms the state-of-the-art systems in terms of computing speed, without compromising the sensitivity or the precision. Our method is between 17 and 40 times faster than the ArUco approach implemented in the OpenCV library. When compared to other approaches such as Chilitags, AprilTags, and ArToolKit+, our method achieves even higher speedups.

We consider as possible future works to investigate the use of the proposed method in fisheye cameras, as well as to characterize the performance when multiple fiducial markers with significantly different scales are present in the same image.

Our system, which is publicly available as open source code[4], is a cost-effective tool for fast and precise self-localization in applications such as robotics, unmanned vehicles and augmented reality applications.

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.

References

[1] R. Sim, J.J. Little, Autonomous vision-based robotic exploration and mapping using hybrid maps and particle filters, Image Vis. Comput. 27 (1) (2009) 167–177. (canadian Robotic Vision 2005 and 2006).
[2] A. Pichler, S.C. Akkaladevi, M. Ikeda, M. Hofmann, M. Plasch, C. Wögerer, G. Fritz, Towards shared autonomy for robotic tasks in manufacturing, Procedia Manuf. 11 (Supplement C) (2017) 72–82. (27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27–30 June 2017, Modena, Italy).
[3] R. Valencia-Garcia, R. Martinez-Béjar, A. Gasparetto, An intelligent framework for simulating robot-assisted surgical operations, Expert Syst. Appl. 28 (3) (2005) 425–433.
[4] A. Broggi, E. Dickmanns, Applications of computer vision to intelligent vehicles, Image Vision Comput. 18 (5) (2000) 365–366.
[5] T. Patterson, S. McClean, P. Morrow, G. Parr, C. Luo, Timely autonomous identification of UAV safe landing zones, Image Vis. Comput. 32 (9) (2014) 568–578.
[6] D. González, J. Pérez, V. Milanés, Parametric-based path generation for automated vehicles at roundabouts, Expert Syst. Appl. 71 (2017) 332–341.
[7] J.L. Sanchez-Lopez, J. Pestana, P. de la Puente, P. Campoy, A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-UAV systems: simulation and experimentation, J. Intell. Robot. Syst. (2015) 1–19.
[8] M. Olivares-Mendez, S. Kannan, H. Voos, Vision based fuzzy control autonomous landing with UAVs: from V-REP to real experiments, Control and Automation (MED), 2015 23th Mediterranean Conference on, 2015. pp. 14–21.
[9] S. Pflugi, R. Vasireddy, T. Lerch, T.M. Ecker, M. Tannast, N. Boemke, K. Siebenrock, G. Zheng, Augmented marker tracking for peri-acetabular osteotomy surgery, 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2017. pp. 937–941.
[10] J.P. Lima, R. Roberto, F. Simões, M. Almeida, L. Figueiredo, J.M. Teixeira, V. Teichrieb, Markerless tracking system for augmented reality in the automotive industry, Expert Syst. Appl. 82 (2017) 100–114.
[11] P. Chen, Z. Peng, D. Li, L. Yang, An improved augmented reality system based on AndAR, J. Vis. Commun. Image Represent. 37 (2016) 63–69. (weakly supervised learning and its applications).
[12] S. Khattak, B. Cowan, I. Chepurna, A. Hogue, A real-time reconstructed 3D environment augmented with virtual objects rendered with correct occlusion, Games Media Entertainment (GEM), 2014 IEEE, 2014. pp. 1–8.
[13] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-Scale Direct Monocular SLAM, 2014.
[14] R. Mur-Artal, J.M.M. Montiel, J.D. Tardós, ORB-SLAM: a versatile and accurate monocular SLAM system, IEEE Trans. Robot. 31 (5) (2015) 1147–1163.
[15] Cooperative pose estimation of a fleet of robots based on interactive points alignment, Expert Syst. Appl. 45 (2016) 150–160.
[16] S.-h. Zhong, Y. Liu, Q.-c. Chen, Visual orientation inhomogeneity based scale-invariant feature transform, Expert Syst. Appl. 42 (13) (2015) 5658–5667.
[17] 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 Recogn. 47 (6) (2014) 2280–2292.
[18] E. Olson, AprilTag: a robust and flexible visual fiducial system, Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011. pp. 3400–3407.
[19] F. Ababsa, M. Mallem, Robust Camera Pose Estimation Using 2D Fiducials Tracking for Real-time Augmented Reality Systems, Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI '04, 2004. pp. 431–435.
[20] 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 Syst. Appl. 93 (1) (2018) 336–345.
[21] R. Muñoz-Salinas, M.J. Marín-Jiménez, E. Yeguas-Bolivar, R. Medina-Carnicer, Mapping and localization from planar markers, Pattern Recogn. 73 (January 2018) 158–171.
[22] K. Dorfmüller, H. Wirth, Real-time Hand and Head Tracking for Virtual Environments Using Infrared Beacons, in Proceedings CAPTECH'98. 1998, Springer. 1998, pp. 113–127.
[23] M. Ribo, A. Pinz, A.L. Fuhrmann, A new optical tracking system for virtual and augmented reality applications, In Proceedings of the IEEE Instrumentation and Measurement Technical Conference, 2001. pp. 1932–1936.
[24] V.A. Knyaz, R.V. Sibiryakov, The Development of New Coded Targets for Automated Point Identification and Non-contact Surface Measurements, 3D Surface Measurements, International Archives of Photogrammetry and Remote Sensing, vol. XXXII, part 5, 1998. pp. 80–85.
[25] L. Naimark, E. Foxlin, Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-inertial Self-tracker, Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 27–36.
[26] J. Rekimoto, Y. Ayatsuka, CyberCode: designing augmented reality environments with visual tags, Proceedings of DARE 2000 on Designing Augmented Reality Environments, DARE '00, ACM, New York, NY, USA, 2000, pp. 1–10.
[27] M. Rohs, B. Gfeller, Using Camera-equipped Mobile Phones for Interacting with Real-world Objects, Advances in Pervasive Computing, 2004. pp. 265–271.
[28] M. Kaltenbrunner, R. Bencina, reacTIVision: a computer-vision framework for table-based tangible interaction, Proceedings of the 1st International Conference on Tangible and Embedded Interaction, TEI '07, ACM, New York, NY, USA, 2007, pp. 69–74.
[29] H. Kato, M. Billinghurst, Marker tracking and HMD calibration for a video-based augmented reality conferencing system, Augmented Reality, 1999. (IWAR '99) Proceedings. 2nd IEEE and ACM International Workshop on, 1999. pp. 85–94.
[30] S. Lin, D.J. Costello, Error Control Coding, Second ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.
[31] D. Wagner, D. Schmalstieg, ARToolKitPlus for Pose Tracking on Mobile Devices, Computer Vision Winter Workshop, 2007. pp. 139–146.
[32] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L.M. Encarnação, M. Gervautz, W. Purgathofer, The Studierstube augmented reality project, Presence Teleop. Virt. 11 (1) (2002) 33–54.
[33] M. Fiala, Designing highly reliable fiducial markers, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2010) 1317–1324.
[34] D. Flohr, J. Fischer, A Lightweight ID-based Extension for Marker Tracking Systems, Eurographics Symposium on Virtual Environments (EGVE) Short Paper Proceedings, 2007. pp. 59–64.
[35] S. Garrido-Jurado, R. Mu noz-Salinas, F. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using mixed integer linear programming, Pattern Recogn. 51 (2016) 481–491.
[36] Q. Bonnard, S. Lemaignan, G. Zufferey, A. Mazzei, S. Cuendet, N. Li, A. Özgür, P. Dillenbourg, Chilitags 2: Robust Fiducial Markers for Augmented Reality and Robotics, 2013, http://chili.epfl.ch/software.
[37] D. Johnston, M. Fleury, A. Downton, A. Clark, Real-time positioning for augmented reality on a custom parallel machine, Image Vis. Comput. 23 (3) (2005) 271–286.
[38] Topological structural analysis of digitized binary images by border following, Comput. Vis. Graph. Image Process. 30 (1) (1985) 32–46.
[39] D.H. Douglas, T.K. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, Cartograph. Int. J. Geogr. Inf. Geovis. 2 (10) (1973) 112–122.
[40] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.
[41] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision in C++ with the OpenCV Library, 2nd ed., O'Reilly Media, Inc. 2013.

[4] http://www.uco.es/grupos/ava/node/25.

Chapter 3

Second contribution. "Fractal Markers: a new approach for long-range camera pose estimation under occlusion"

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 21, 2019, accepted October 31, 2019, date of publication November 4, 2019, date of current version December 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951204

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

FRANCISCO J. ROMERO-RAMIRE[1], RAFAEL MUÑOZ-SALINAS[1,2], AND R. MEDINA-CARNICER[1,2]

[1]Departamento de Informática y Analisis Numérico, Edificio Einstein. Campus de Rabanales, Universidad de Córdoba, 14071 Córdoba, Spain
[2]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 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

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

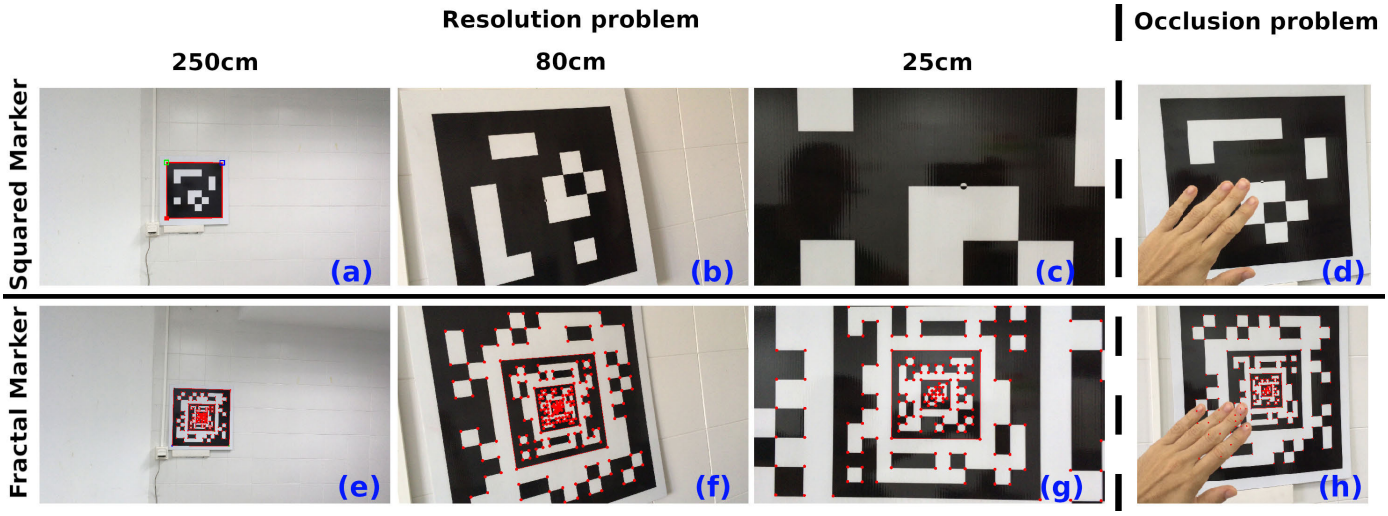


FIGURE 1. Common problems of squared markers: the resolution problem (a-c) and the occlusion problem (d). Fig. (a-c) show a squared marker observed at the distances 250 *cm*, 80 *cm* and 25 *cm* from the camera and overlaid as red rectangles the ArUco [4] detections (only works in the first case). Under 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 submarkers. 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 $(f^1, f^2, \ldots, f^m)$, 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.

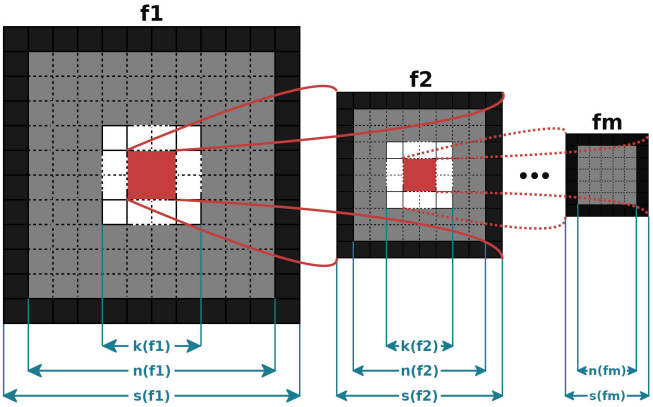


FIGURE 2. Generic structure of Fractal Marker F, in which each marker is composed of a set of cells that can be grouped into three categories. The black band correspond to the marker border, the gray cells configure and uniquely determine the marker, and finally, the white band facilitate the detection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

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 $f^i, i \in \{1, \ldots, m\}$ is (see Fig. 2):

$$S_R(f^i) = n(f^i)^2 - k(f^i)^2. \quad (1)$$

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 $s(f^3) = 4$, $n(f^3) = 2$, $k(f^3) = 0$, $S_R(f^3) = 4$.

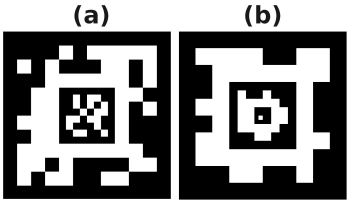


FIGURE 3. Examples of different configurations of Fractal Marker and areas of identification region $S_R(f^i)$. (a) Fractal Marker composed of two internal markers $F = \{f^1, f^2\}$, whose identification areas are $S_R(f^1) = 64$ and $S_R(f^2) = 36$. (b) Fractal Marker composed of three internal markers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

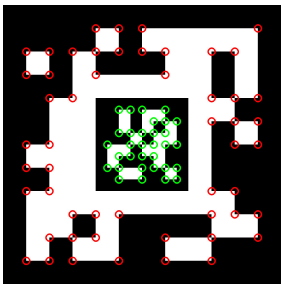


FIGURE 4. Fractal Marker composed of two internal markers. The inner corners of marker f^1 and f^2 are shown in red and in green respectively.

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

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

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., $b_j^i \sim Be(1/2)$). 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.:

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

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:

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

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. 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-dimentional 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

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

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 Marker F.

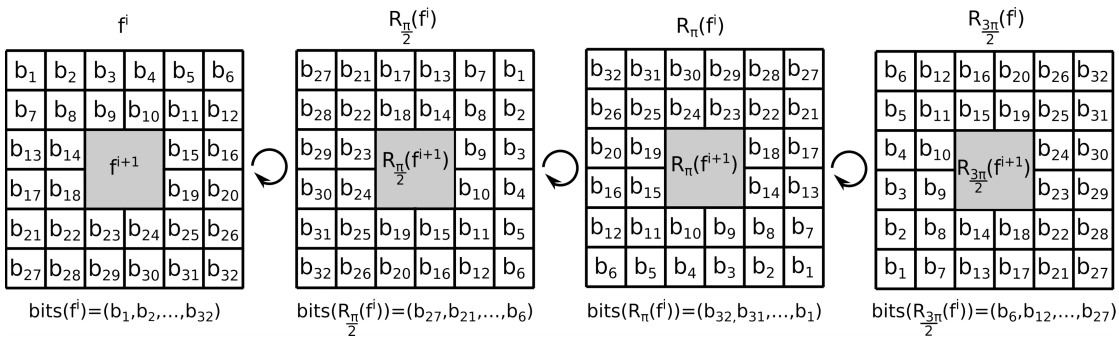


FIGURE 5. Four possible rotations of a marker f^i.

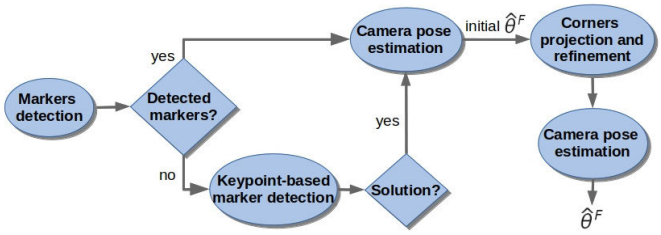


FIGURE 6. General workflow of proposed method for marker pose estimation.

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 and 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 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 [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, 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).

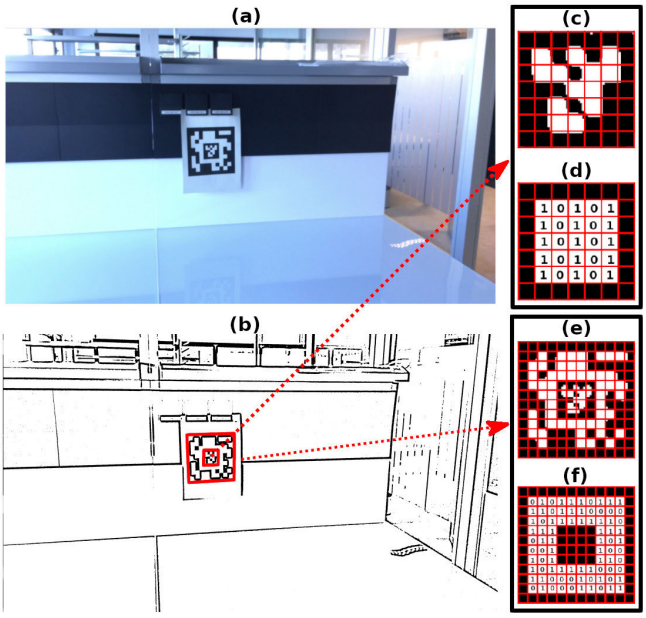


FIGURE 7. Detection and identification of Fractal Markers. (a) Original image. (b) Thresholded image showing the result of contour extraction and filtering. (c and e) Canonical images of rectangular contours containing our markers. (d and f) Binarized versions of the canonical images.

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

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

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

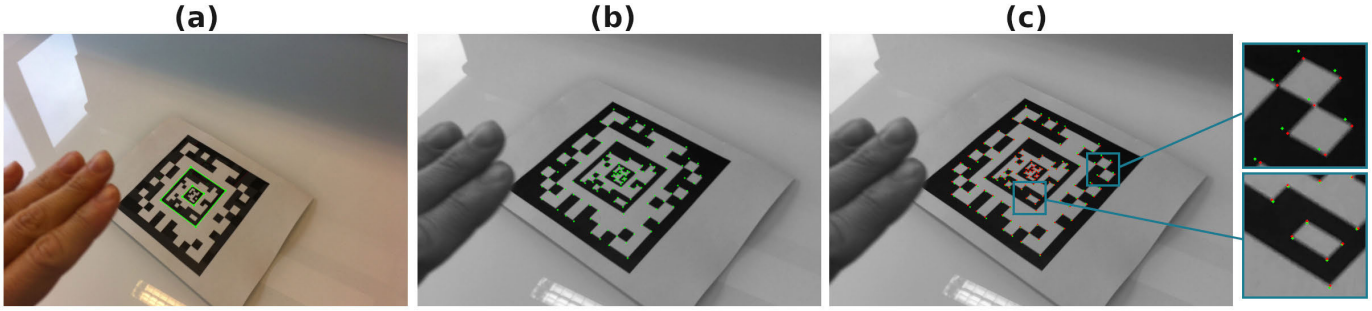


FIGURE 8. (a) Detection of markers and external corners in original image. (b) Initial estimation of the position using external corners of the detected markers. (c) Refinement of the pose estimation: the green points represent the estimate of the previous step (b), in red the new estimation.

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

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

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

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

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

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 $\mathcal{I} = (I^0, I^2, \ldots, I^p)$ 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:

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

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., $P(f) < \tau(f)$), 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 τ_c, the corner is considered occluded and discarded from the process. Second, we discard corners that undergo large displacements during the refinement process.

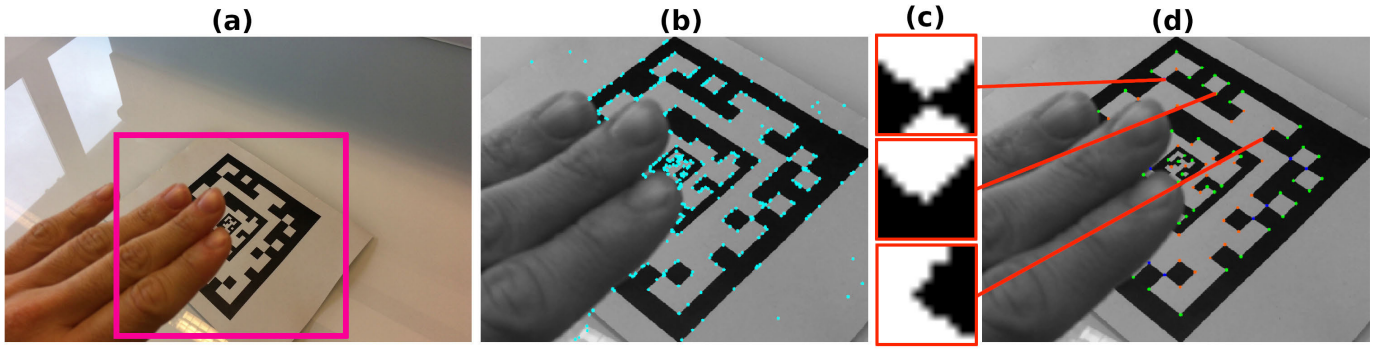


FIGURE 9. (a) Original image showing the region of interest. (b) Results of applying the FAST detector (blue dots). (c) Examples of corner classification (d) Filtered and classified keypoints. Each color (blue, green and red dots) represent a different keypoint class.

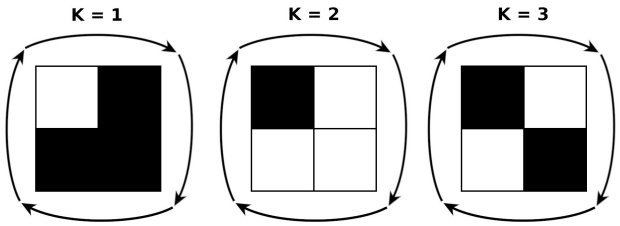


FIGURE 10. The three categories a keypoint can belong to. Each keypoint will be assigned to one of these three categories, or discarded.

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 τ_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: $R \leftarrow roi(I, k, l)$ *# 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. 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 × 2160 and all the images and videos employed for experiments are publicly available.[1] 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.

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).
α	0.7	Percentage of matches needed to consider a generated model as good (Section IV-D).
β	0.1	Percentage of minimum matches necessary to consider the model as a candidate (Section IV-D).
τ_c	25	Contrast threshold for corners (Sections IV-C and IV-D).

A. DETECTION RANGE ANALYSIS

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

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

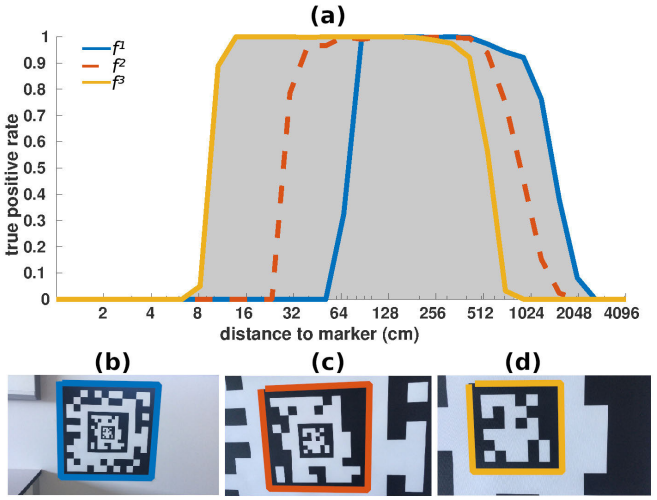


FIGURE 11. (a) True positive detection rates as a function of the distance to the markers. Each coloured line correspond to one of the inner markers that compose the Fractal Marker. The grey area correspond to the detection range of the complete Fractal Marker. (b-d) Different views of the Fractal Marker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

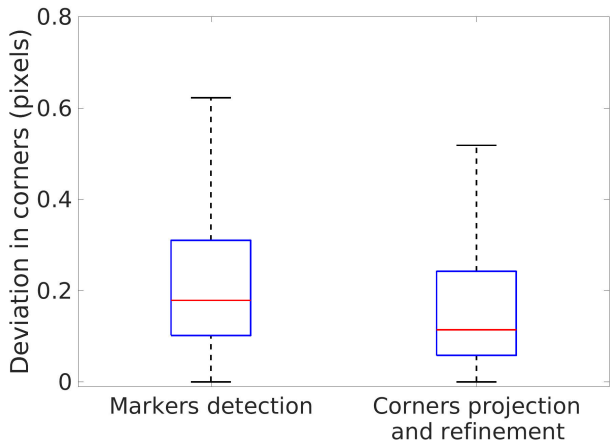


FIGURE 12. Vertex jitter before and after the proposed corner refinement. The proposed method improves accuracy.

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 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.

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

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

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 $\times$ 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

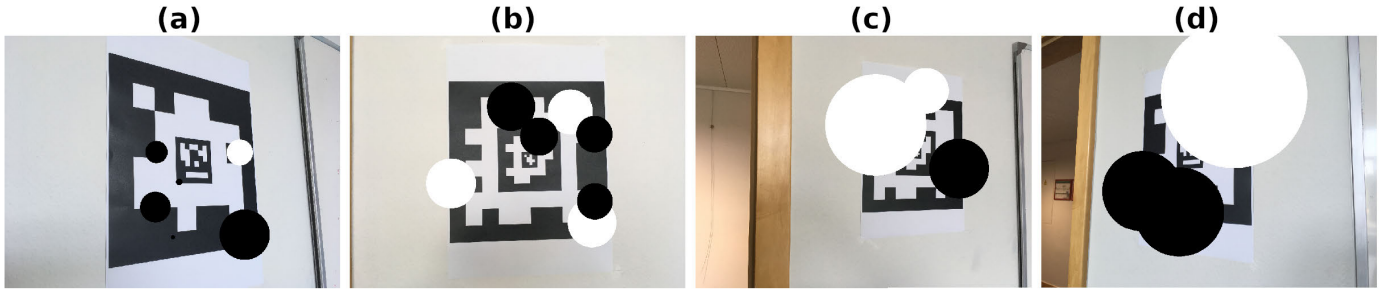


FIGURE 13. Some of the images employed to test detection under occlusion. Different levels of occlusion are synthetically added to the images: (a) 11.29%, (b) 33.19%, (c) 53.92%, (d) 73.37%.

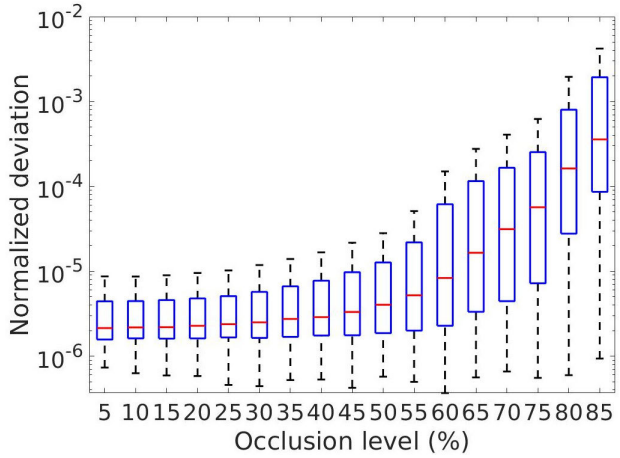


FIGURE 14. Average (red) and Standard deviation (blue) of the normalized error for different occlusion levels. See text for details.

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 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.

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.

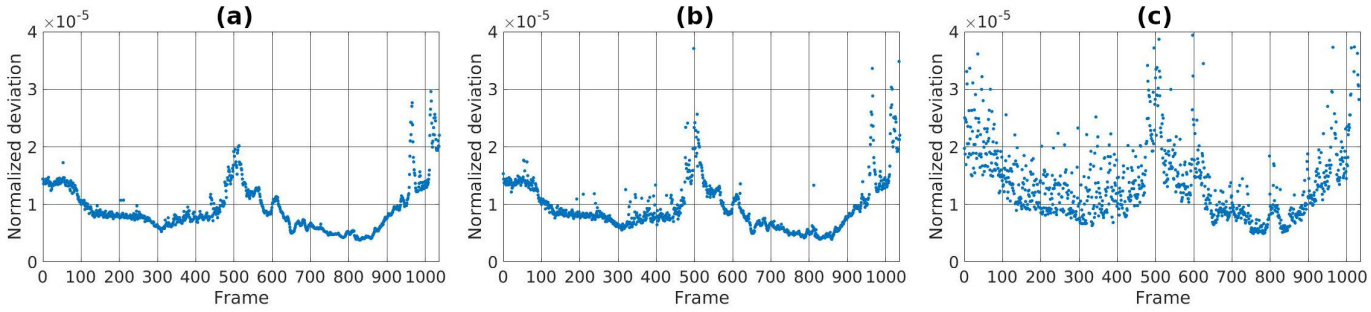


FIGURE 15. Normalized pixel error of the Keypoint-based marker detection method for one video sequence using different levels of synthetic occlusion: (a) 0%, (b) 30%, (c) 60%.

VI. CONCLUSION

This paper has proposed the Fractal Marker, a novel type of marker that can be detected in a wider range of distances 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,[2] and is publicly available for other researchers to use it.

As possible future work, we point out the possibility of generating multiple Fractal Makers 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.

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

[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. Mallem, "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 peri-acetabular 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. Shastry, 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 quadrocopter using fuzzy control and ArUco markers," *Intell. Service Robot.*, vol. 10, pp. 185–194, Jul. 2017.

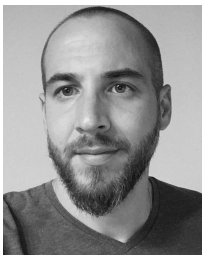
[16] H. Sarmadi, R. Muñoz-Salinas, M. Á. Berbís, A. Luna, and R. Medina-Carnicer, "3D reconstruction and alignment by consumer RGB-D sensors and fiducial planar markers for patient positioning in radiation therapy," *Comput. Methods Programs Biomed.*, vol. 180, Oct. 2019, Art. no. 105004.

[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.

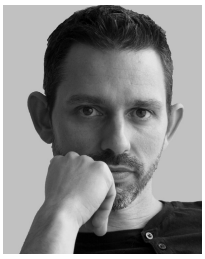
[18] M. Behroozi, A. Lui, I. Moore, D. Ford, and C. Parnin, "Dazed: Measuring the cognitive load of solving technical interview problems at the white-board," in *Proc. IEEE/ACM 40th Int. Conf. Softw. Eng., New Ideas Emerg. Technol. Results (ICSE-NIER)*, May 2018, pp. 93–96.

[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.

[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. Haggenmiller, 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 Analisis 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.; Orebro, 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 Cordoba (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.

Chapter 4

Third contribution. "Tracking fiducial markers with Discriminative Correlation Filters"

Image and Vision Computing 107 (2021) 104094

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Tracking fiducial markers with discriminative correlation filters

Francisco J. Romero-Ramirez [a], Rafael Muñoz-Salinas [a,b,*], Rafael Medina-Carnicer [a,b]

[a] Departamento de Informática y Análisis Numérico, Edificio Einstein, Campus de Rabanales, Universidad de Córdoba, 14071 Córdoba, Spain
[b] Instituto Maimónides de Investigación en Biomedicina (IMIBIC), Avenida Menéndez Pidal s/n, 14004 Córdoba, Spain

ARTICLE INFO

Article history:
Received 26 July 2020
Received in revised form 10 December 2020
Accepted 11 December 2020
Available online 31 December 2020

Keywords:
Discriminative correlation filter
Squared fiducial markers
Marker mapping SLAM

ABSTRACT

In the last few years, squared fiducial markers have become a popular and efficient tool to solve monocular localization and tracking problems at a very low cost. Nevertheless, marker detection is affected by noise and blur: small camera movements may cause image blurriness that prevents marker detection.

The contribution of this paper is two-fold. First, it proposes a novel approach for estimating the location of markers in images using a set of Discriminative Correlation Filters (DCF). The proposed method outperforms state-of-the-art methods for marker detection and standard DCFs in terms of speed, precision, and sensitivity. Our method is robust to blur and scales very well with image resolution, obtaining more than 200fps in HD images using a single CPU thread.

As a second contribution, this paper proposes a method for camera localization with marker maps employing a predictive approach to detect visible markers with high precision, speed, and robustness to blurriness. The method has been compared to the state-of-the-art SLAM methods obtaining, better accuracy, sensitivity, and speed. The proposed approach is publicly available as part of the ArUco library.

1. Introduction

Squared fiducial markers have become a popular and efficient method to solve monocular localization and tracking problems at a very low cost in indoor environments. In medical applications, they are used for tracking of surgical equipment [1–3]. In augmented reality (AR) problems, it is employed to estimate the camera pose so as to properly render the scene [4,5]. In autonomous navigation or drone landing, it provides visual references for navigation and landing [6–8].

The recent works in squared markers [9,10] make it is possible to estimate the camera pose in the environment (with the correct scale) by just analyzing images where some markers are visible. Given a set of these markers printed on a regular piece of paper and placed randomly in the environment (Fig. 1a), it is possible to estimate their three-dimensional location from a set of images or a video sequence showing them (Fig. 1b). This method allows obtaining motion tracking systems of very low cost, requiring only a camera.

Nevertheless, one of the limitations of these techniques is that the detection of markers is sensitive to blurring. Fig. 2 shows the appearance of the markers under different blurring levels obtained by moving the camera at different speeds. Even at low camera speeds, manually recorded videos have blurriness that prevents detection (see Fig. 2b). This effect happens either because the camera, which is not placed on a gimbal (e.g. in low-cost AR applications or drone landing), or because the marker moves fast (surgical equipment tracking). The high sensitivity to blurring is a limitation to the spread of that technology in applications of low cost and low computing power.

The contribution of this paper is two-fold. First, this work proposes a novel approach for estimating the location of markers in images, that is both fast and robust to blur, which consists in employing a set of Discriminative Correlation Filters (DCF). In order to speed up computation, our method employs a pyramid of images and selects at each frame the one where tracking can be done at maximum speed. Fig. 2 shows the tracking capabilities of the proposed method. As a second contribution, we propose a novel approach for monocular camera pose estimation using marker maps. The proposed method, given a marker map, employs the previous trackers and a predictive approach to detect visible markers with high precision, speed, and robustness to blurriness.

The experiments conducted shows that the proposed marker tracking method is fastest and more robust to blur than the state-of-the-art marker detection algorithms, and more precise than the best DCFs. In addition, our proposal is compared with three state-of-the-art SLAM methods: ORBSlam2 [11], LDSO [12], and UcoSLAM [13]. Our method outperforms them in terms of speed and precision. The proposed method is publicly available as part of the ArUco library.[1]

* Corresponding author at: Departamento de Informática y Análisis Numérico, Edificio Einstein, Campus de Rabanales, Universidad de Córdoba, 14071 Córdoba, Spain.
E-mail addresses: fj.romero@uco.es (F.J. Romero-Ramirez), in1musar@uco.es (R. Muñoz-Salinas), rmedina@uco.es (R. Medina-Carnicer).

[1] https://www.uco.es/investiga/grupos/ava/node/26

https://doi.org/10.1016/j.imavis.2020.104094
0262-8856/

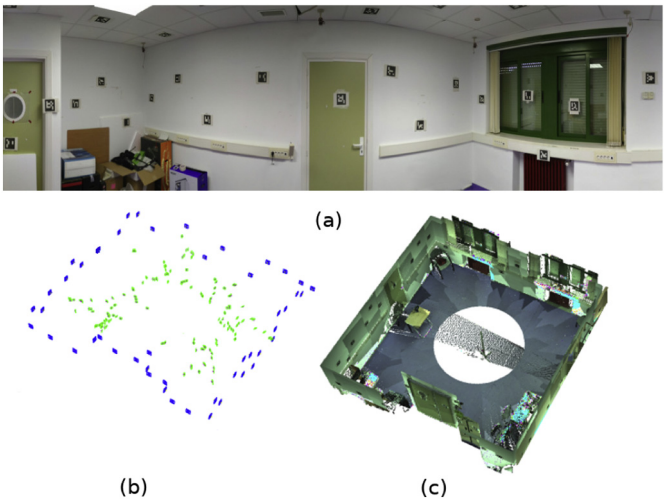


Fig. 1. Map of markers generated by the works [9,10]. (a) Image of tracking room where a set of markers are randomly placed in the walls. (b) Marker map generated with [9]. Blue squares represents the pose of the markers and green ones the pose of the cameras. (c) Laser reconstruction of the room to help to understand the three-dimensional configuration of the room.

The remainder of this paper is structured as follows. Sect. 2 provides an overview of the related works, while Sect. 3 explains the basis of DCF. Our contributions are explained in Sect. 4 and 5, while Sect. 6 presents the experiments conducted and Sect. 7 draws some conclusions.

2. Related works

2.1. Fiducial marker systems

A large number of systems based on planar markers have emerged due to their high accuracy, robustness, and speed in camera pose estimation and tracking processes.

Some works propose the use of circular planar markers [14–17], where the identification is embedded in sectors or concentric rings. Also, FourierTag [18] where the information is represented in the frequency domain and the length of information provided is a function of the distance from the camera to the marker.

However, systems based on square planar markers are the most widely used [10,19–24].

ChromaTag [23], unlike traditional gray planar markers, proposes the use of colored markers where extreme color gradients are used in the initial marker detection, reducing the number of candidate markers to be analyzed.

To solve the camera pose estimation problem, some works have focused on the use of fiducial markers in the Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) methods. These marker-based systems solve several problems associated with keypoint-based approaches, such as scale, rotations, or areas with low texture (corridors, ceilings, etc.). The MarkerMap [25] work proposes a method for the creation of a marker map, solving the problem of ambiguity. On the other hand, work [26] presents an SfM method where markers are used to correct image matches. The recent work UcoSLAM [13] combines the use of natural and artificial landmarks. The system allows obtaining the scale of the map as soon as a marker has been detected in the environment and does a continuous localization by fusing information from natural landmarks and markers, which provide stable references along time.

2.2. Discriminative correlation filters

Since the appearance of the work of Bolme et al. [27] with Minimum Output Sum of Squared Error (MOSSE), discriminative correlations filters (DCF) have increased their popularity, becoming one of the main methods of visual tracking due to its efficiency and robustness.

Many other researchers have work on improving several aspects of the initial MOSSE proposal. Henriques et al. [28] replaces the use of grayscale filters by using HOG features, Danelljan et al. [29] introduces learning multi-channel filters with Colornames, Li et al. [30] and Lukežič et al. [31] use the integration of both HOG and Colornames. Other works employing convolutional features of CNNs [32–34] have shown high performance.

DCF usually has limited information about the contour, leading to false positives in some scenarios such as rapid movement, occlusion, or background noise. Mueller et al. [35] use context information in filter training to improve the performance of state-of-art algorithms without incurring in high computational costs. On the other hand, to reduce the boundary effects Danneljan et al. [36] reformulate the learning function by considering larger image regions, penalizing filter values outside the bounding box.

Another limitation of DCFs is the assumption that the target has a fixed size and that it is completely aligned to a rectangular region. However, the shape of the tracked objects and their rotation makes the filter learn the background, leading to errors in tracking. Danelljan et al. [37] presented a method to estimate the scale by training a classifier on a pyramidal scale. Also, Lukežič et al. [31] introduce the channel and spatial reliability concepts. The spatial reliability map adjusts the filter to the

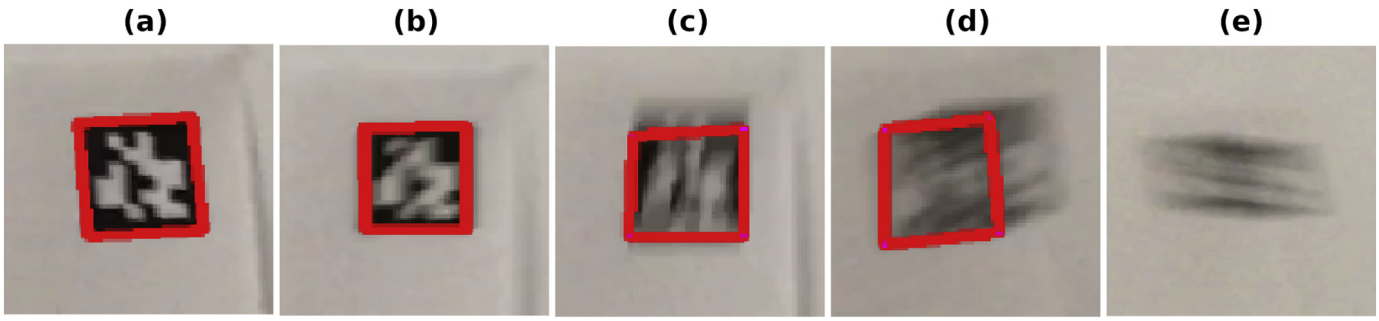


Fig. 2. Tracking of a fiducial marker along the a video sequence with the proposed method. From left to right, the marker is observed with increasing blurring levels. The proposed method is capable of tracking the marker in figures a-d but not in figure e. The estimated marker location is drawn as a red rectangle.

object to the object allowing to adapt the size of the search region and improving the tracking of non-rectangular objects. The channel reliability reflects the discriminative power of each filter channel.

3. Mathematical basis of discriminative correlation filters

Correlation filter based tracking applies a continuous adaptive process to find the filter that when applied on the desired target produces the maximum response. In its simpler form, the filter is a small image patch centered around the object to be tracked. However, in order to increase the robustness to appearance changes, a set of modified images of the target (created using affine transformations) are employed to build the filter. Once the initial filter is created, the filter is applied on the next image at the same location. Then, the position with the maximum response within the region is considered the new target location. Finally, the filter is updated to adapt changes in appearance and the process repeated in the subsequent frames [27].

Let us denote $\mathcal{X} = \{x_1, \ldots, x_n\}$ the set of gray-scale patches of the target observed under different appearance conditions. It will be used as a training set to create the initial filter h. Also, let us denote $\mathcal{G} = \{g_1, \ldots, g_n\}$ the desired response of the filter when applied on the patches, i.e., $h(x_i) = g_i$. Although g_i can have any shape, it is generally generated as a 2D Gaussian ($\sigma = 2$) centered at the center of the patch. Thus, in practice $g_i = g_j \; \forall \; i, j \in \{1,\ldots,n\}$.

Computing the correlation in the Fourier Domain has demonstrated to be the best way to speed up computation and obtain a certain degree of robustness to misalignment. Correlation in the frequency domain turns into element-wise multiplications expressed as:

$$G = X \odot H^* \tag{1}$$

where G, X and H denotes the Fourier transforms of $g \in \mathcal{G}, x \in \mathcal{X}$ and h respectively, $\odot$ is the element-wise multiplication and $*$ the complex conjugate. In consequence, the estimation of the optimal correlation filter H in the Fourier domain is computed as:

$$\min_H \sum_{i=1}^{n} \|X_i \odot H^* - G_i\|^2 + \lambda_1 \|H\|^2 \tag{2}$$

where λ_1 is a regularization term. Since Eq. 2 is convex, it has a single global minimum that can be expressed as:

$$H_1 = \frac{\sum_{i=1}^{n} X_i \odot G_i^*}{\lambda_1 + \sum_{i=1}^{n} X_i \odot X_i^*} \tag{3}$$

which expresses how to obtain the correlation filter in the first frame. The regularization term λ_1 prevents divisions by zero.

In frame t ($t > 1$), the filter is applied to the previous target location and the location with maximum response is expected to be current target location. We shall define z_t as the image patch centred at the maximum response location in t, and Z_t is its Fourier transform. Then, the filter is updated using a running average so that

$$H_t = \frac{\eta A_t + (1-\eta) A_{t-1}}{\eta B_t + (1-\eta) B_{t-1}} \tag{4}$$

$$A_t = G_t \odot Z_t^* \tag{5}$$

$$B_t = Z_t \odot Z_t^* \tag{6}$$

where the parameter $\eta \in [0,1]$ is the learning rate.

An important aspect to consider is how to detect when the tracking has failed. A method to do so is analyzing the Peak to Sidelobe Ratio (PSR), which is the ratio between the filter value at the point with maximum response and the average response in the rest of the pixels. It has been observed than values below 7.0 indicates tracking failure [27].

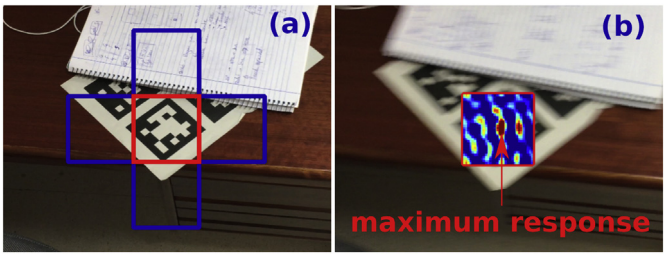


Fig. 3. Tracking process with correlative filters. (a) Training process in frame t. The filter is updated using the central patch of the marker in addition to the 4 patches around it. (b) represents the process of tracking in the frame t+1, for it uses the filter updated in t, the maximum value of response indicates the new position of the marker.

In general, the area surrounding the tracked object may contain distracting information for tracking that leads to an erroneous local minimum. An effective approach to alleviate this problem is to include contextual information in the filter [35]. Instead of considering only the target appearance to build the filter, patches surrounding the target are also employed as negative examples. Following this approach, Eq. 2 is updated so that the minimization function takes into account a set of patches surrounding the target.

If $\mathcal{Y} = \{y_1, \ldots, y_m\}$ is the set of contextual patches (blue patches in Fig. 3a), then Eq.2 becomes:

$$\min_H \sum_{i=1}^{n} \|X_i \odot H^* - G_i\|^2 + \lambda_1 \|H\|^2 + \lambda_2 \sum_{j=1}^{m} \|Y_j \odot H\|^2 \tag{7}$$

where λ_2 modulates the relative importance of the context and Y_j is the Fourier transform of y_j. Using this approach, the update of the filter in frame t ($t > 1$) is expressed as:

$$H_t = \frac{\eta A_t + (1-\eta) A_{t-1}}{(\eta B_t + (1-\eta) B_{t-1}) + \lambda_2 (\eta D_t + (1-\eta) D_{t-1})} \tag{8}$$

where

$$D_t = \sum_{i=1}^{m} Y_{i,t} \odot Y_{i,t}^* \tag{9}$$

and A_t, B_t are obtained from Eqs. 5 and 6.

4. Tracking of a squared marker

This section introduces our first contribution, a DCF-based tracker that allows the continuous tracking of a square fiducial marker throughout a video sequence. Since estimating the exact location of the marker corners is required to estimate its three-dimensional pose, our method must be able to track them. Therefore, our approach employs a total of five gray-scale filters: one filter for tracking the marker general appearance, and four additional filters for tracking the corners. In order to speed up computation while adapting to scale changes, a multi-resolution pyramid tracking approach is proposed. Filters of fixed size are employed (constraining the computation time), but, at each iteration, the scale where the marker dimension best fits the filter size is calculated as later explained in Eq. 11.

Our process can be summarized in the following steps. In the first frame, we find the pyramid level where the filters are created with the desired size. In subsequent frames, the filters are first applied in the neighboring regions of the previous location at the same pyramid level to find the optimal location of the marker and its corners. Then,

to adapt to scale changes of the marker, we must find the scale that produces the highest response of the filter. Finally, the filters are updated.

Below, we provide a formal description of the proposed method.

4.1. Tracker definition and initialization

The initial step to track a marker $\mathbf{m}$ along a video sequence is to find it in the image. A marker is a squared matrix in which each element represents a bit (see Fig. 4). The marker is comprised of a black region, which helps to detect it, and the inner region containing the bits that uniquely identify the marker. Let us define the sequence of bits of a marker as

$$b(\mathbf{m}) = (b_1, \ldots, b_n) \mid b_i \in \{0, 1\}, \tag{10}$$

which is created row by row starting at the top-left bit of the matrix. The detection of the marker in the image can be efficiently done using the method proposed in [10]. The method extracts contours in the image, obtain its polygonal approximation, and discard those that are not quadrilateral. Each remaining polygon $\mathbf{p}$ is analyzed to check if it belongs to a valid marker. Its four corners are employed to compute Homography matrix that determines the central pixel of each bit in the image and its pixel intensities are thresholded using Otsu's [38] algorithm obtaining its bit sequence $b(\mathbf{p})$ in its four main rotations (0°, 90°, 180° and 270°). If the Hamming distance of both is zero in any of the possible rotations, then we have a perfect match and the marker is considered as detected (see Fig. 4).

Let us define

$$c = \left\{ c_k \mid c_k \in \mathrm{R}^2, k \in \{1, \ldots, 4\} \right\}$$

as the pixel coordinates of the four corners for marker $\mathbf{m}$ in image $\mathbf{I}$, $\mathcal{C}(\mathbf{m}) \in \mathrm{R}^2$ as the location of the marker center, and $\mathcal{A}(\mathbf{m})$ as the observed marker area.

Our aim is to use patches of length side τ_s to create the DCFs for marker and corners. To do so, the patches are obtained from an down-sampled version of the image $\mathbf{I}$ where the marker area $\mathcal{A}(\mathbf{m})$ is most similar to τ_s^2. If we denote

$$\mathrm{J} = \left(I^0, I^1, \ldots, I^n \right)$$

as the pyramid of images ($I^0 = \mathbf{I}$) where the image $I^j, j > 0$ is the original image $\mathbf{I}$ down-sampled by the factor

$$\beta^j \mid \beta \in [0, 1],$$

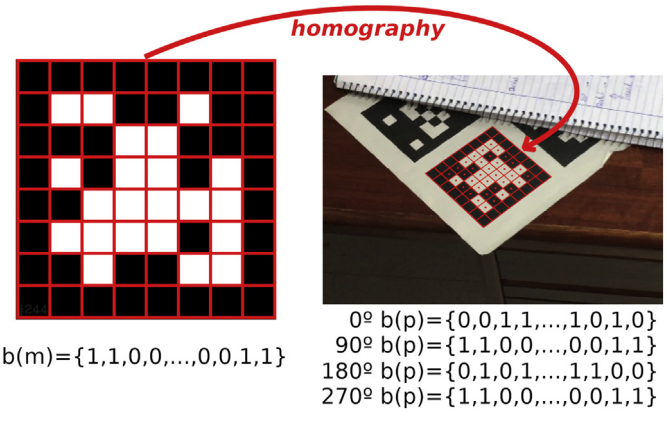


Fig. 4. Identification of the tracked marker. The computation of the homography on the detected polygon, allows to take the central value of its identification bits, and analyzed in its four possible orientations.

then, we can define:

$$L(\mathbf{m}) = \begin{cases} 0 & \text{if } \left(\dfrac{\tau_s^2}{A(\mathbf{m})} \right) \geq 1 \\ \left\lfloor \log_\beta \left(\dfrac{\tau_s^2}{A(\mathbf{m})} \right) \right\rfloor & \text{otherwise} \end{cases} \tag{11}$$

as the pyramid level where the area $\mathcal{A}(\mathbf{m})$ of the marker is most similar to the desired patch area τ_s^2. In other words, the image $I^{L(\mathbf{m})}$ is where the initial patches of area τ_s^2 will be extracted. Please notice that $\lfloor \cdot \rfloor$ denotes the floor function.

We shall define $P(p, \tau_s)$ as the function that returns a patch of size τ_s^2 centred at $p \in \mathbb{R}^2$ in the image $I^{L(\mathbf{m})}$. Consequently, the patches to generate the DCFs for the marker and its corners are $P(\mathcal{C}(\mathbf{m}), \tau_s), P(c_1, \tau_s), P(c_2, \tau_s), P(c_3, \tau_s)$ and $P(c_4, \tau_s)$, respectively.

Let us then define the tracker for marker $\mathbf{m}$ at time t as:

$$T_t^{\mathbf{m}} = \{ T_{0,t}^{\mathbf{m}}, \ldots, T_{4,t}^{\mathbf{m}}, l_t^{\mathbf{m}} \} \tag{12}$$

where $T_{i,t}^{\mathbf{m}}$ represents the Fourier transforms of the DCF for the marker center ($T_{0,t}^{\mathbf{m}}$) and its four corners ($T_{i,t}^{\mathbf{m}}, i \in \{1, \ldots, 4\}$) (see Eq. 7), while $l_t^{\mathbf{m}}$ represents the pyramid level employed for correlation at time t. In the first frame,

$$l_1^{\mathbf{m}} = L(\mathbf{m}).$$

4.2. Tracking and update

In subsequent frames ($t > 1$), the filters are applied at the previous location, and the location of maximum filter response is obtained:

$$\mathscr{E}_{i,t}^{\mathbf{m}}(l_t^{\mathbf{m}}) = \underset{p \in \mathbb{R}^2}{\operatorname{argmax}} \, PSR\left(T_{i,t}^{\mathbf{m}}, p, l_t^{\mathbf{m}} \right), \tag{13}$$

where PSR indicates the response of the filter $T_{i,t}^{\mathbf{m}}$ centred at pixel p in the image $I^{l_t^{\mathbf{m}}}$. If the maximum PSR for the marker tracker $T_{0,t}^{\mathbf{m}}$ is below the established threshold value, the marker is considered as lost.

A very important aspect to consider is the need for an accurate estimation of the marker corners. The corner locations estimated by Eq. 13 do not have the required accuracy for pose estimation. First, because tracking normally is done at a reduced version of the original image. Second, even if the tracker is run at the lowest piramid level I^0, the result is not accurate enough. The corners locations must be refined with sub-pixel accuracy. Thus, in order to obtain a precise corner estimation, we employ an iterative corner upsampling process that produces a precise corner location $\mathcal{S}(T_{i,t}^{\mathbf{m}})$ in the original image I^0. To do so, first, a corner search with sub-pixel accuracy is performed in the vicinity of the estimated corner locations $\mathscr{E}_{i,t}^{\mathbf{m}}(l_t^{\mathbf{m}})$. For that purpose, the subpixel refinement method described in [39] is employed. Then, the corner location is upsampled to the previous pyramid level $l_t^{\mathbf{m}} - 1$, and the search repeated. The process stops when the image I^0 is reached.

Adapting to scale is another crucial element for a successful tracking. In the first frame, correlation is done at the pyramid level $l_1^{\mathbf{m}}$ where the DCFs were initialized. However, due to scale changes of the marker (when approaching or moving away from it), the initial pyramid level $l_1^{\mathbf{m}}$ may not be the one for which the filters obtain its maximum response. Thus, it is necessary to find the best pyramid level for the next frame. To do so, the response of the filter $T_{0,t}^{\mathbf{m}}$ at the contiguous pyramid scales is analyzed, and the one maximizing the marker filter is selected:

$$l_{t+1}^{\mathbf{m}} = \underset{l \in \{l_t^{\mathbf{m}}+1, l_t^{\mathbf{m}}, l_t^{\mathbf{m}}-1\}}{\operatorname{argmax}} PSR(T_{0,t}^{\mathbf{m}}, \mathrm{E}_{0,t}^{\mathbf{m}}(l), l) \tag{14}$$

Once the best pyramid level is found, all the filters are updated using the patches extracted from that level.

4.3. Confidence measure

The proposed method can track the marker $\mathbf{m}$ under large appearance changes caused by blur (see Fig. 2). However, in some cases, the blurriness level is so high that the estimated location of the corners is not reliable enough for three-dimensional pose estimation.

We propose a confidence measure $w^{\mathbf{m}} \in [0,1]$ that indicates how reliable is the estimation provided by our tracker. As it will become evident in the next section, this measure will allow favoring some markers over others when doing localization from multiple makers. Values near to 1 indicate high confidence in the detection while values near to zero indicate low confidence.

The measure is composed of two terms. First, the normalized Hamming H distance between the marker bit sequence $b(\mathbf{m})$ and the bit sequence $b(\mathbf{p})$ observed for the polygon $\mathbf{p}$ formed by the four marker corners estimated by our tracker:

$$\frac{\mathrm{H}(b(\mathbf{m}), b(\mathbf{p}))}{|b(\mathbf{m})|}.$$

But then, this value is modulated by the response of the corner trackers

$$\frac{\sum_{i=1}^{4} \mathcal{PSR}_i}{4},$$

where

$$\mathcal{PS}.\mathscr{R}_i = \begin{cases} 1 & PSR\left(T_{i,t}^{\mathbf{m}}, \mathscr{E}_{i,t}^{\mathbf{m}}(l), l\right) > \chi \\ 0 & otherwise \end{cases} \tag{15}$$

indicates if the tracking of a corner was successful or not.

Thus, the confidence measure is expressed as:

$$w^{\mathbf{m}} = 1 - \left(\frac{\mathrm{H}(b(\mathbf{m}), b(\mathbf{p}))}{|b(\mathbf{m})|} \frac{\sum_{i=1}^{4} \mathcal{PSR}_i}{4} \right). \tag{16}$$

We have found after several experiments that the combination of both terms provides better results than any one of them separately.

5. Robust marker-map based pose estimation

This section explains our second contribution, an extension of the previous methodology aimed at camera pose estimation with marker maps. A marker map is a set of markers placed in known map locations of the environment that are employed for camera localization in indoor environments. The observation of a single marker can be enough to obtain the pose of the camera on the map. However, the more markers are visible, the better the accuracy that can be obtained (see Fig. 7).

Our goal is estimating the camera pose $\theta_t \in \mathbb{R}^6$ (position and angle) in the map given: (i) a set of markers $\mathscr{M}$ in known map locations, (ii) an image $\mathbf{I}_t$ showing some of them, and (iii) the previous camera location θ_{t-1}.

We shall define the set of markers in our map by

$$\mathrm{M} = \{\mathbf{m} = \{q_1^{\mathbf{m}}, \ldots, q_4^{\mathbf{m}}\}\}, \tag{17}$$

where $q_i^{\mathbf{m}} \in \mathbb{R}^3$ represents the three-dimensional coordinates of the marker corners in the environment. The map can be obtained from images of the environment using any of the methods described in [9,13,25].

Given an image showing some of the markers, it is possible to estimate the camera pose by analyzing the set of 2D-3D correspondences. Since the 3D location of the corners is known in advance ($\mathscr{M}$), their 2D image projections can be employed to find the pose between the camera and the global reference system by minimizing the reprojection of the observed markers as will be explained later in Sect. 5.2.

The rest of this Section explains the proposed method to estimate the camera pose θ_t given an input image $\mathbf{I}_t$, which can be summarized in Alg 5.1.

5.1. Method overview

Our method employs a set of trackers

$$\mathcal{T}_t = \{T_t^{\mathbf{m}}\}, \mathbf{m} \in \mathrm{M}, t \geq 1,$$

to estimate the position of the markers in the image, where $T_t^{\mathbf{m}}$ is the type of tracker defined in the previous Section (Eq. 12). We are proposing a tracking method, and thus, it requires to be initialized. The initial position θ_1 and $\mathcal{T}_1$ are obtained from the markers detected with a marker detector [22,40,41].

In subsequent frames $\mathbf{I}_t$, the trackers $\mathcal{T}_t$ are applied in order to find the new markers locations. Tracking of a marker may fail for several reasons: it fall outside the image view, occlusion, high blur, etc. Thus, we remove from $\mathcal{T}_t$ the trackers $T_t^{\mathbf{m}}$ with a low response (PSR) of the central tracker $T_{0,t}^{\mathbf{m}}$. The corners of the remaining markers are employed to obtain an initial estimation of the camera pose $\hat{\theta}_t$ (Sect. 5.2).

As the camera moves along the environment, some markers will fall out of the camera view while others will appear. Since we know both the pose of the camera $\hat{\theta}_t$ and the three-dimensional location of the markers $\mathscr{M}$, we can estimate which markers should be visible in the current image and where (Sect. 5.3). For each expected visible marker, (not in $\mathcal{T}_t$) a quick detection is done on the expected image region where it should be visible. If correctly detected, a new tracker $T_t^{\mathbf{m}}$ is added to $\mathcal{T}_t$. After all the new markers have been added, we calculate the final camera pose θ_t using all the visible markers.

Tracking may fail either because of very fast movement causing a lot of blur (Fig. 2e), or because there are no visible markers in the image. Thus, as final step we analyze if a tracking confidence measure $w^{\mathcal{T}_t}$ (explained in Sect. 5.4) is high enough. If not, the tracking should stop until a reliable pose can be obtained using a regular marker detector [22,40,41] to restart tracking.

Algorithm 1 Tracking algorithm overview for image $\mathbf{I}_t$

Algorithm 1: Tracking algorithm overview for image $\mathbf{I}_t$

```
Data: M, θ_{t-1}, T_{t-1}, I_t
Result: θ_t, T_t, w^{T_t}
begin
    T_t ⟵ ApplyFilters(T_{t-1}) (Sect 4);
    for T_t^m ∈ T_t do
        if PSR(T_t^m) < χ then
            remove T_t^m from T_t
        end
    end
    Estimate pose θ̂_t using T_t (Sect. 5.2);
    Look for new visible marker (Sect. 5.3);
    for each new maker m do
        add T_t^m to T_t
    end
    Obtain the final pose θ_t using updated T_t;
    Calculate tracking confidence w^{T_t} (Sect. 5.4);
end
```

5.2. Camera pose estimation

The estimation of the camera pose given a set of markers detected in the image consists in minimizing the reprojection error of their corners, considering its confidence $w^{\mathbf{m}}$ (Eq. 16):

$$\theta_t = \underset{\theta}{\operatorname{argmin}} \sum_{\mathbf{m} \in \mathscr{M}} w^{\mathbf{m}} \cdot \mathscr{H}(e_t^{\mathbf{m}}(\theta)), \tag{18}$$

where $e_t^{\mathbf{m}}(\theta)$ represents the reprojection error of the corners of marker $\mathbf{m}$ and $\mathscr{H}$ is the Hubber function, employed to minimize the impact of possible outliers:

$$\mathrm{H}(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \alpha \\ \alpha\left(|a| - \frac{1}{2}\alpha\right) & \text{otherwise} \end{cases} \tag{19}$$

The reprojection error of a marker $\mathbf{m}$ is defined as:

$$e_t^{\mathbf{m}}(\theta) = \sum_{i=1}^{4} \|\psi(\mathrm{q}_\mathrm{i}^{\mathbf{m}}, \theta) - \mathcal{S}(\mathrm{T}_{\mathrm{i,t}}^{\mathbf{m}})\|^2, \tag{20}$$

where the function $\psi(q,\theta) \in \mathbb{R}^2$ projects the three-dimensional point q in the image given the camera pose θ and $\mathcal{S}(T_{i,t}^{\mathbf{m}})$ is the precise corner location in the original image I^0 (section 4.2).

Eq. 18 is a non-linear function that can be efficiently minimized using the Levenberg–Marquardt's (LM) algorithm [42].

5.3. Look for visible markers

As the video sequence progresses, and the camera moves, markers will appear and disappear from the scene. The initialization of these markers is essential to achieve continuous tracking and accurate pose estimation.

Given that the three-dimensional locations of the marker corners in $\mathscr{M}$ are known, and an initial camera pose $\widehat{\theta}_t$ for the $\mathbf{I}_t$ image is available, we can calculate which markers should be visible in the $\mathbf{I}_t$ image and where their corner should project.

For each expected marker, we apply a detection process in the region where it should be visible. First, the Otsu's thresholding algorithm [38] is applied, and contours are extracted using the Suzuki and Abe algorithm [43]. Using the Douglas and Peucker algorithm [44], the largest a squared polygon $\mathbf{p}$ is selected. Then, the bits $b(\mathbf{p})$ of the polygon are extracted and if they match the predicted marker $b(\mathbf{m})$, using the Hamming distance, the marker is considered found and a tracker initialized and added to $\mathcal{T}_t$ to be employed for the next image.

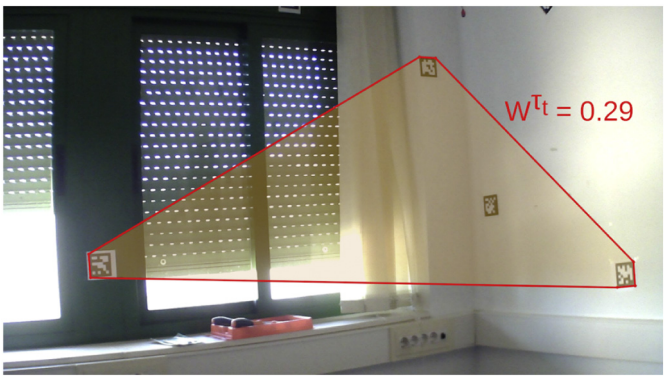


Fig. 5. The confidence value $w^{\mathcal{T}_t}$ in the tracking is given by the hull convex area obtained from the vertices of the detected markers in the image.

Table 1
Nomenclature and values of the main parameters used by the proposed method.

Parameter	Default value	Description
λ_1	10^{-4}	Filter regulation parameter (Sect 3)
λ_2	20	Context-Aware parameter (Sect 3)
η	0.2	Learning rate (Sect 3)
τ_s	32	Filter size (Sect 4.1)
β	0.7	Pyramid scale factor (Sect 4.1)
χ	5.7	Peak to Sidelobe Ratio (Alg. 5.1)
α	2.5	Hubber function cut-off value (Sect 5.2)
τ_c	0.1	Tracking Confidence Threshold (Sect 5.4)

5.4. Calculate tracking confidence

As previously mentioned, tracking may fail due to the absence of markers, blur, bad lightning conditions, marker occlusion or any other reason. Therefore, it is important to provide a confidence value indicating how reliable the estimated pose θ_t is. It allows determining whether tracking has failed, and in that case, the system can stop tracking and use a slower but more conservative method for detecting the markers [22,40,41].

In this paper, we propose a confidence measure based on the following principle. If a single marker is spotted very near to the camera, occupying a large region of the image, the estimation of the pose is reliable. However, if the same marker is detected far from the camera, occupying only a very small region of the image, the estimation is very unreliable. In the end, the reliability of the estimated pose depends mainly on the total area of the points employed for computing Eq. 18. If the points are far apart, occupying a large region of the image, the estimated pose is reliable, and vice versa. So, let us define the confidence measure $w^{\mathcal{T}_t}$ as the area of the convex hull formed by the marker corners employed in Eq. 18, divided by the total image area (see Fig. 5). This value is one of the points cover all the image, and tends to zero as they are more concentrated in a region. If the confidence $w^{\mathcal{T}_t}$ is below a threshold τ_c, we consider the tracking has failed.

6. Experiments and results

This section explains the experiments carried out to validate our proposal. The goal of the experiments is to evaluate the robustness, speed, and accuracy of the proposed method for marker tracking. Experiments have been divided in two categories. First, the individual marker tracking algorithm (Sect. 4) is tested, comparing it with state-of-the-art marker detection methods (Sect. 6.1), and correlation filter trackers (Sect. 6.2). Afterward, our method for camera pose estimation using marker maps (Sect. 5) is compared with the state-of-the-art SLAM methods in challenging video sequences (Sect. 6.3).

All experiments have been performed using an Intel® processor Core™ i7-7500 U CPU @ 2.70GHz × 4, with 8Gb of Ram, and the Ubuntu 18.04 operating system. Although some of the processing could be parallelized, only one thread has been used.

Several parameters that control the behavior of the proposed algorithms we have introduced along the paper. The values used for these parameters have been experimentally selected and are shown in Table 1.

Finally, we must indicate that the code has been integrated as part of the public library ArUco. We will refer to the proposed method as TR-ArUco. The code and the videos recorded to conduct the experiments are publicly available.[2]

[2] https://www.uco.es/investiga/grupos/ava/node/69

6.1. Comparison with fiducial squared marker detectors

This section makes an analysis of the effect of the motion blur in terms of detection rate and speed of the proposed method TR-ArUco against the main state-of-the-art marker detection and tracking algorithms: ArUco [40] and AprilTag [22]. Nowadays, both methods are widely used due to their high performance in terms of speed detecting fiducial markers.

Both AprilTag and ArUco detector has different configurable parameters establishing a balance between speed and detection range. For the AprilTag detector, this parameter is the decimation factor, and for the ArUco detector, it is the minMarkerSize. To do a fair comparison, parameter values that maximize the number of detections are chosen. Thus, for AprilTag the decimate factor 2 has been employed, and the minMarkerSize is set to 0 for ArUco. Additionally, for the ArUco method two versions have been used: the *ArUco_NORMAL* detection method, which employs an adaptive image threshold, and the *ArUco_FAST* detection method that uses a global threshold.

A set of video sequences have been recorded showing a squared marker (of size 6×6 cm) printed on a piece of paper. Along the sequences, the marker remains static, while the camera moves at different speeds and distances from the marker. In total 6043 video frames, of resolution 1920×1080, have been recorded using a mobile phone. The marker has been recorded according to an ideal non-blur scenario, with smooth movements between frames so that the marker is always detected by all methods. From the original video sequences (labeled as blur0), we have simulated four linear blurring levels (blur3, blur6, blur9, and blur12), using the motion blur filter of the Gimp software.[3]

We have analyzed four parameters for each method: (i) the error in the estimation of the marker corners (expressed in pixels), (ii) precision, (iii) recall, and (iv) processing speed. Table 2 shows the results of the experiment for the different levels of motion blur. Since no method has reported a false positive, the precision is 1 for all the methods. Thus, we have not reported that value in the Table. On the other hand, our method obtains the highest recall for all levels of blur tested, and this is specially noticeable for levels 9 and 12. It is mainly at these levels that the edges of the marker becomes very diffuse (see Fig. 2(c-e)), making it difficult to accurately estimate the corners. For this reason, our method, which detects in such a difficult situation, obtains higher errors than the rest of the methods in that test. Finally, we must mention that our method is faster than the other methods.

Table 3 shows a summary of the average time employed by the different steps of the TR-ArUco method, for different image resolutions (namely 1080p, 720p and 480p). Notice that steps $1.1 - 1.2$ are only performed when the marker is not being tracked, i.e. in the first frame of the video sequence, or when a marker that was being tracked is lost. Steps $2.1 - 2.4$ are performed on all frames. As can be seen, the computation times for the resolutions used are similar, with an average computation time of 4.19 ms. Among the different phases, the selection of the optimal scale is the most time-consuming one.

6.2. Comparison with discriminative correlation filters

This Section compares the proposed method with the state-of-the-art Discriminative Correlation Filters trackers, namely, KCF [45], CSRT [31], MIL [46], TLD [47], MEDIANFLOW [48], MOSSE [27] and BOOSTING [49]. The implementations provided in the public library OpenCV[4] library have been employed.

The key aspect when detecting a squared fiducial marker is correctly detecting the position of its four corners in the image. Consequently, this experiment aims at evaluating the capability of the above indicated DCFs to track the four corners of a marker.

For this experiment, a total of 10 video sequences have been recorded using a mobile phone, containing a total of 3326 video frames of resolution 1920×1080. The videos show a marker recorded from different directions, orientations, and motion speeds: fast movements (i.e. with blur) are followed by moments of no movement. While the trackers will be fed with all the video frames, only these without blur are considered for evaluation, because estimating the corner location in the blurred images is not accurate. Please look at Fig. 2d and try to select the correct corner locations in the image. In consequence, while tracking errors/failures of the tested methods will mostly occur in the blurred frames, their consequences will be measured later, when the image is stabilized.

Then, instead of manually annotating the location of the marker corners, we have employed the ArUco marker detector, which only detects the marker in the images without blur, but achieving subpixel accuracy. Therefore, only the frames in which the marker is detected with ArUco (i.e. no blur), are considered for numerical evaluation.

For each one of the selected DCFs trackers, we have applied the following methodology. A total of four independent trackers have been employed to track the marker corners. The trackers are initialized in the first frame to the center of each corner, and then, the trackers are applied to the subsequent frames. The size of the filters is half the size of the marker (see Fig. 6). Whenever the tracking error becomes higher than a number of pixels ε, the trackers are initialized so as to avoid the trackers to become completely lost for the rest of the sequence. For our tracker, we proceed in a similar way, re-initializing the tracker if the error in the estimation of the corners becomes greater than ε.

The results obtained for different values of ε are shown in Table 4, where each row represents a method. The columns show the total number of re-initialization required in the sequences evaluated (init), the average frames per second employed by each method (fps), and the average tracking error (err) which is expressed in pixels. In this set of experiments, only images of resolution 1080*p* have been employed.

As can be observed, the proposed method TR-ArUco outperforms the rest of the methods in the three parameters evaluated. Our method obtains a stable frame rate, which is an order of magnitude faster than the rest of the methods (except for MOSSE). The same can be said about the number of re-initializations, which is much lower than in the rest of the algorithms. Finally, the tracking error of the corners of our method is the lowest of all. The main conclusion is that the proposed method outperforms the naive approach (i.e., using individual DCFs) for the given problem.

Fig. 2 shows some of the images evaluated in this experiment, overlaying in red the estimations obtained. Fig. 2-(e) shows a case in which our method fails and requires re-initialization. As can be seen, our method requires re-initialization only in very extreme cases.

6.3. Comparison with SLAM methods

This section analyzes the TR-ArUco method for camera pose estimation using marker maps (Sec. 5), with the state-of-art SLAM methods. The following SLAM algorithms have been tested:

- ORBSlam2 [11]: a SLAM method based on keypoints.
- LDSO [12]: a SLAM method based on photoconsistency.
- *ArUco_MM* [25]: a SLAM method based on fiducial squared markers.
- UcoSLAM [13]: a SLAM method using both keypoints and fiducial squared markers.

For evaluation purposes we have employed two different datasets: the publicly available SPM dataset [9], and a new dataset created for this paper (the DCF dataset[5]).

Both datasets have been recorded in our laboratory where a set of fiducial squared markers have been placed at random locations. The SPM

[3] https://www.gimp.org/
[4] https://opencv.org/
[5] https://mega.nz/folder/LiRCDYYb#aAOjirkUt54-0CGr3C6-1g

Table 2
Results of tested methods for different blur levels. See text for details.

	FPS	*blur*0		*blur*3		*blur*6		*blur*9		*blur*12	
		err	Recall	err	Recall	err	Recall	err	Recall	err	Recall
TR-ArUco	**234.796**	0.37	**1.0**	0.76	0.99	1.7	**0.97**	2.74	**0.88**	3.47	**0.68**
AprilTag	26.978	0.83	**1.0**	0.79	**1.0**	**1.22**	0.96	**2.01**	0.71	2.91	0.44
ArUco_NORMAL	102.737	**0.0**	**1.0**	**0.67**	0.96	1.84	0.71	2.27	0.23	3.09	0.03
ArUco_FAST	198.292	**0.0**	**1.0**	**0.67**	0.97	1.82	0.76	3.14	0.4	**2.02**	0.19

Table 3
Mean computing times (milliseconds) of the different steps of the proposed method for different resolutions.

	Resolution		
	480p	720p	1080p
Step 1.1:ArUco detect	0.204	0.704	0.947
Step 1.2:Creating filters	0.028	0.029	0.031
Time Step 1 (ms)	0.232	0.733	0.978
Step 2.1:Convert to gray	0.267	0.492	1.365
Step 2.2:Optimal scale	1.191	1.382	1.645
Step 2.3:Track corners	0.984	1.006	1.038
Step 3.3:Track marker	1.050	1.111	1.142
Time Step 2 (ms)	3.492	3.991	5.190

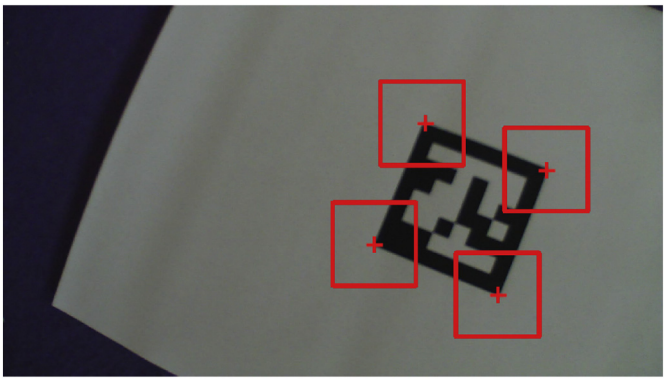


Fig. 6. Naive approach employed to track a marker consist in using four independent DCFs: one for each corner.

dataset consists of eight video sequences recorded with a PtGrey FLEA3 camera capturing 1920 × 1080 images at 60 Hz. The videos show up to fifty different fiducial markers of 16.5 cm, distributed in the walls and ceiling of the room. The DCF dataset has nine video sequences recorded with an ELP camera capturing at 30 Hz frame rate with a resolution of 1920 × 1080 pixels. In this case, a total of 102 markers of a smaller size (7.9 cm), have been distributed by the walls and ceiling of the room. The videos of the DCF dataset have been recorded moving the camera fast and with brusque movements with the aim of achieving different degrees of blurring. In both cases, the ground truth camera poses are obtained using an Optitrack motion capture system equipped with six cameras (see Fig. 7).

While the ORBSlam2 and LDSO makes no use of the markers explicitly, the *ArUco_MM* and UcoSLAM methods use the markers for tracking. However, our method, TR-ArUco, requires the location of the marker to be know in advance (i.e. the marker map). The map has been created with the UcoSLAM method using a long video sequence that covers all markers in the room.

For the SLAM methods, the following methodology has been employed to analyze the video sequences. The sequence has been first processed to obtain the map and then, using the generated map, it is processed again to estimate the camera poses at each frame. In this way, the SLAM methods are evaluated after correcting possible loops in the sequence and obtains better accuracy. In consequence, a fair comparison with our method, that has a known map of the environment build in a previous phase, can be made.

Table 5 show the results obtained. For each video sequence (row) and method (column), three measures have been obtained. First, the computing time (FPS). Second, the Absolute Trajectory Error (ATE), which is the translational RMSE after *Sim*(3) alignment [50] of the estimated poses with the ground truth. And third, the percentage of the video sequence frames for which the method provides a pose estimation (%Trck). It must be indicated that SLAM systems do not provide estimations in all the frames of a sequence: in some cases, they get lost due to fast movement or lack of texture.

Two conclusions can be drawn from Table 5. First, the proposed method outperforms the others in terms of speed and percentage of tracked frames. Second, that the LDSO method performs poorly in most of the sequences tested.

However, comparing the results of two SLAM methods is not a trivial task. Imagine a method that only estimates the pose of the camera in the first ten frames while a second method estimates poses in the whole

Table 4
Results obtained by different state-of-the-art DCF trackers. We evaluate the total number of tracking re-initializations (init), the computation time (fps), and the average tracking error (err).

	$\varepsilon < 5$			$\varepsilon < 10$			$\varepsilon < 15$		
	init	fps	err	init	fps	err	init	fps	err
TR-ARUCO	**24**	287.42	**0.82**	**18**	266.14	**0.98**	**13**	294.24	**1.13**
CSRT	72	7.12	1.63	60	8.15	1.76	57	8.35	1.85
BOOSTING	98	13.16	1.83	91	12.68	1.92	84	12.85	2.30
MEDIANFLOW	116	19.55	1.92	77	25.26	2.92	61	21.26	4.06
MIL	245	5.99	2.68	163	6.00	3.97	112	5.90	4.65
MOSSE	270	**1105.06**	2.31	214	**1035.57**	3.38	166	**1020.56**	4.20
KCF	338	91.29	2.74	236	79.82	4.97	186	75.23	6.73
TLD	734	4.90	4.08	539	2.37	7.45	409	2.39	10.48

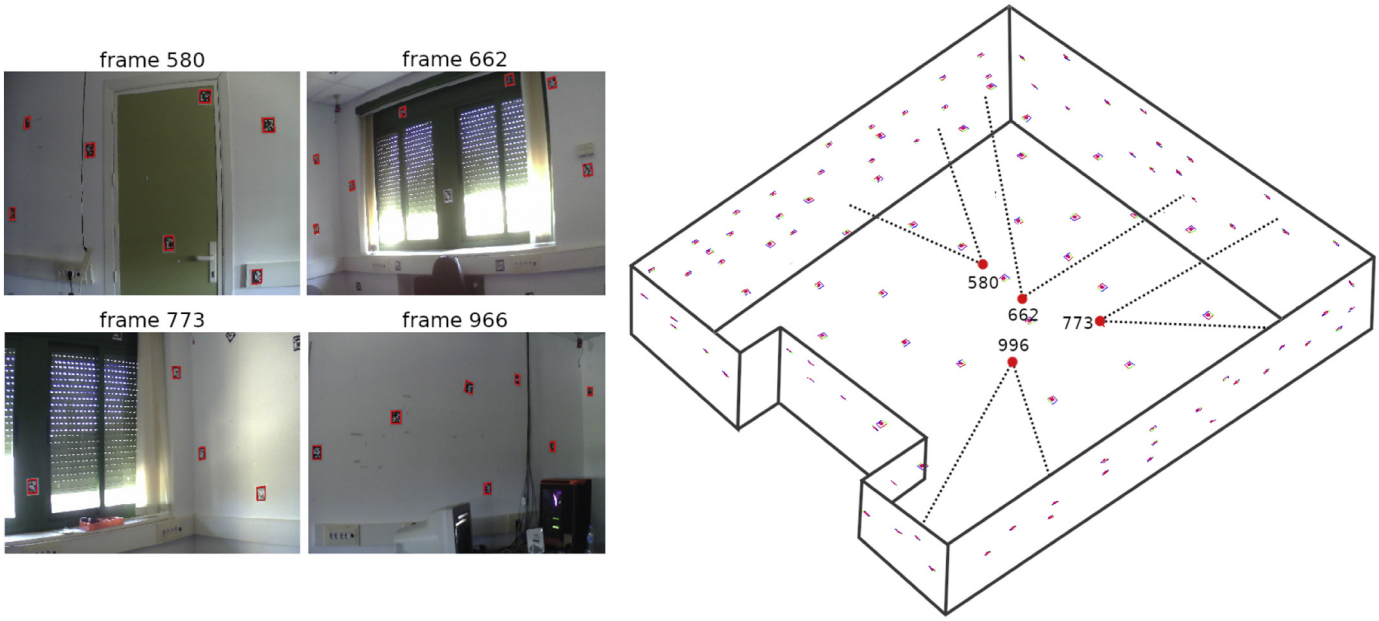


Fig. 7. Map of markers displayed in the laboratory for experimentation. Some scenes of the environment corresponding to the first video are shown in it.

Table 5
Results obtained for each method in the SMP [9] and DCF datasets. For each sequence, the frames per second (FPS), absolute trajectory error (ATE), and percentage of tracked frames (%Trck) are reported.

Dataset	Sequence	TR-ArUco			ArUco_MM			LDSO			ORB_SLAM2			UcoSLAM		
		FPS	ATE	%Trck	FPS	ATE	%Trck	FPS	ATE	%Trck	FPS	ATE	%Trck	FPS	ATE	%Trck
SPM	video1	**58.5**	0.068	**99.8**	48.8	**0.062**	99.6	2.97	0.769	46.2	12.6	2.360	99.3	9.96	0.378	65.1
SPM	video2	**62.0**	0.111	**99.8**	53.4	0.103	98.5	1.47	1.250	**99.8**	10.8	0.575	97.4	22.1	**0.054**	**99.8**
SPM	video3	**49.1**	0.061	**99.8**	42.8	0.058	98.2	1.65	2.320	**99.8**	12.6	**0.054**	**99.8**	24.7	0.098	**99.8**
SPM	video4	44.6	0.015	**99.8**	**45.5**	0.013	99.2	0	∞	0.03	12.2	0.020	**99.8**	24.9	**0.011**	**99.8**
SPM	video5	38.8	0.023	**99.8**	**41.4**	**0.019**	98.6	0	∞	0.04	11.8	1.410	94.7	23.1	0.026	98.0
SPM	video6	34.2	0.145	**99.8**	**45.1**	**0.018**	98.6	0	∞	0.04	11.4	0.527	96.8	6.46	0.670	52.2
SPM	video7	**36.0**	**0.950**	99.8	31.3	1.020	99.2	0	∞	0.05	9.62	1.280	99.4	17.3	1.860	**100.**
SPM	video8	36.5	0.077	**99.9**	**41.8**	0.077	99.6	0	∞	**0**	3.05	0.437	55.0	17.9	**0.049**	99.8
DCF	video1	**44.2**	0.116	**97.4**	27.5	0.108	73.6	0	∞	**0**	1.11	0.499	30.4	5.46	**0.095**	57.6
DCF	video2	**43.6**	**0.095**	**96.5**	31.1	0.114	80.2	0	∞	**0**	0	∞	**0**	9.18	0.109	73.1
DCF	video3	**51.7**	0.085	**99.9**	42.6	**0.082**	91.9	0	∞	**0**	0	∞	**0**	12.3	0.105	87.6
DCF	video4	**46.5**	**0.072**	**99.9**	44.2	0.076	93.2	0	∞	**0**	0	∞	**0**	12.7	0.074	88.8
DCF	video5	**29.0**	0.163	**81.9**	19.4	0.106	60.3	0	∞	**0**	1.56	0.293	38.0	4.07	**0.081**	53.7
DCF	video6	**38.7**	0.093	**99.9**	31.6	0.101	82.3	0	∞	**0**	0	∞	**0**	7.82	**0.092**	75.8
DCF	video7	**42.8**	0.116	**94.7**	27.6	0.114	72.3	0	∞	**0**	0.04	**0.040**	6.62	5.61	0.102	65.0
DCF	video8	52.1	0.067	**99.9**	**52.6**	0.071	98.5	0	∞	**0**	7.39	0.303	84.4	14.8	**0.065**	96.6
DCF	video9	41.5	0.074	**99.9**	**45.2**	0.082	96.0	0	∞	**0**	0	∞	**0**	11.0	**0.067**	88.5

sequence. Because of the reduced drift in the first frames, the total ATE of the first method will be smaller than the ATE of the second method (which evaluates the whole sequence). This is why (%Trck) is also an important aspect to consider.

The work [13] proposes an evaluation methodology to compare two SLAM methods A and B combining both the ATE and the %Trck. It defines a measure $S_p(A,B) \in [-1,1]$ that employs a confidence level $p \in [0,1]$. When $S_p(A,B)$ is close to 1, it indicates that the A method is better than B, while values close to -1 indicates that the B method is better than A. Table 6 shows the values of $S_p(A,B)$ for each pair of methods, using the 17 sequences of the SPM and DCF datasets, for different confidence values. As can be seen, the proposed method TR-ArUco obtains best scores than the rest of the methods for different confidence levels. The last row of the Table indicates how many times a method obtains better results than other methods. In our case, the value 4 means that proposed method wins to the other four tested methods.

The main conclusion that can be obtained from this experiment is that the proposed method outperforms the state-of-the-art SLAM methods in terms of speed, accuracy and sensitivity, for this particular problem.

7. Conclusions

This paper has proposed methods for tracking squared fiducial markers under challenging conditions. Our first contribution is a method for tracking squared marker using a set of Discriminative Correlation Filters which combines a proper scale selection and a corner upsampling strategy. The proposed method outperforms state-of-the-art methods for marker detection and standard DCFs in terms of speed, precision and sensitivity. In addition, our method scales very well with image resolution, obtaining more than 200fps in HD images using a single CPU thread.

Table 6
Measure $S_p(A,B)$ according to different confidence levels p of the analyzed methods. The final ranking shows TR-ArUco as the best, while LDSO provides the worst scores.

method A / method B	TR-ArUco			ArUco_MM			UcoSLAM			ORB_SLAM2			LDSO		
	$p = 0.01$	$p = 0.1$	$p = 0.25$	$p = 0.01$	$p = 0.1$	$p = 0.25$	$p = 0.01$	$p = 0.1$	$p = 0.25$	$p = 0.01$	$p = 0.1$	$p = 0.25$	$p = 0.01$	$p = 0.1$	$p = 0.25$
TR-ArUco	–	–	–	−0.5	−0.21	−0.15	−0.56	−0.5	−0.29	−0.75	−0.62	−0.58	−0.25	−0.25	−0.25
ArUco_MM	0.5	0.21	0.15	–	–	–	−0.26	−0.21	−0.18	−0.62	−0.58	−0.58	−0.12	−0.25	−0.25
UcoSLAM	0.56	0.5	0.29	0.26	0.21	0.18	–	–	–	−0.38	−0.29	−0.25	−0.25	−0.25	−0.25
ORB_SLAM2	0.75	0.62	0.58	0.62	0.58	0.58	0.38	0.29	0.25	–	–	–	−0.062	−0.12	−0.12
LDSO	0.25	0.25	0.25	0.12	0.25	0.25	0.25	0.25	0.25	0.062	0.12	0.12	–	–	–
Times winner	**4**	**4**	**4**	3	3	3	2	2	2	1	1	1	0	0	0

Our second contribution is a method for low-cost camera pose estimation using fiducial marker maps. The proposed method is able to estimate the pose of a camera by tracking the position of the already visible markers and predicting the location of the markers appearing in the scene. Our method has been compared to state-of-the-art SLAM methods obtaining, better accuracy, sensitivity, and speed.

The proposed methods are publicly available for other researchers as part of the ArUco library,[6] and the datasets employed in this paper are available to ease the reproduction of the experiments. As future work, we consider using the proposed method for map marker creation.

Credit author statement

This is the work of a PhD student and its two supervisors. Francisco is the PhD, Rafael Muñoz is the main supervisor, and Rafael Medina is the cosupervisor and Full Profesor, head of the Research Group.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project has been funded under projects TIN2019-75279-P and IFI16/00033 (ISCIII) of Spain Ministry of Economy, Industry and Competitiveness, and FEDER.

References

[1] H. Nakawala, G. Ferrigno, E.D. Momi, Development of an intelligent surgical training system for thoracentesis, Artif. Intell. Med. 84 (2018) 50–63.
[2] P. Matthies, B. Frisch, J. Vogel, T. Lasser, M. Friebe, N. Navab, Inside-Out Tracking for Flexible Hand-held Nuclear Tomographic Imaging, IEEE Nuclear Science Symposium and Medical Imaging Conference, USA, San Diego, 2015.
[3] P.K. Kanithi, J. Chatterjee, D. Sheet, Immersive augmented reality system for assisting needle positioning during ultrasound guided intervention, in: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP '16, ACM, New York, NY, USA, 2016 65:1–65:8.
[4] E. Marchand, H. Uchiyama, F. Spindler, Pose estimation for augmented reality: A hands-on survey, IEEE Trans. Vis. Comput. Graph. 22 (12) (2016) 2633–2651.
[5] H. duan, Q. Zhang, Visual measurement in simulation environment for vision-based uav autonomous aerial refueling, IEEE Trans. Instrum. Meas. 64 (9) (2015) 2468–2480.
[6] A. Marut, K. Wojtowicz, K. Falkowski, Aruco markers pose estimation in uav landing aid system, 2019 IEEE 5th International Workshop on Metrology for AeroSpace (MetroAeroSpace) 2019, pp. 261–266.
[7] M.F. Sani, G. Karimian, Automatic navigation and landing of an indoor ar. drone quadrotor using aruco marker and inertial sensors, in: 2017 International Conference on Computer and Drone Applications (IConDA), 2017 102–107.
[8] R. Polvara, S. Sharma, J. Wan, A. Manning, R. Sutton, Towards autonomous landing on a moving vessel through fiducial markers, 2017 European Conference on Mobile Robots (ECMR) 2017, pp. 1–6.
[9] R. Muñoz-Salinas, M.J. Marín-Jiménez, R. Medina-Carnicer, Spm-slam: Simultaneous localization and mapping with squared planar markers, Pattern Recogn. 86 (2019) 156–171.
[10] 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 Recogn. 47 (6) (2014) 2280–2292.
[11] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (5) (2017) 1255–1262.
[12] X. Gao, R. Wang, N. Demmel, D. Cremers, Ldso: Direct sparse odometry with loop closure, 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018, pp. 2198–2204.
[13] R. Muñoz-Salinas, R. Medina-Carnicer, Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers, Pattern Recogn. 101 (2020) 107193.
[14] V.A. Knyaz, R.V. Sibiryakov, The development of new coded targets for automated point identification and non-contact surface measurements, 3D Surface Measurements, International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, part 5 1998, pp. 80–85.

[6] https://www.uco.es/investiga/grupos/ava/node/26

[15] L. Naimark, E. Foxlin, Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker, Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, IEEE Computer Society, Washington, DC, USA 2002, pp. 27–36.
[16] L. Calvet, P. Gurdjos, C. Griwodz, S. Gasparini, Detection and Accurate Localization of Circular Fiducials under Highly Challenging Conditions, Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, United States 2016, pp. 562–570.
[17] F. Bergamasco, A. Albarelli, E. Rodola, A. Torsello, Rune-tag: A high accuracy fiducial marker with strong occlusion resilience, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on 2011, pp. 113–120.
[18] J. Sattar, E. Bourque, P. Giguère, G. Dudek, Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction, Fourth Canadian Conference on Computer and Robot Vision (CRV '07) 2007, pp. 165–174.
[19] 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 85–94.
[20] Q. Bonnard, S. Lemaignan, G. Zufferey, A. Mazzei, S. Cuendet, N. Li, A. Özgür, P. Dillenbourg, Chilitags 2: Robust fiducial markers for augmented reality and robotics, http://chili.epfl.ch/software 2013.
[21] D. Wagner, D. Schmalstieg, ARToolKitPlus for pose tracking on mobile devices, Computer Vision Winter Workshop 2007, pp. 139–146.
[22] E. Olson, Apriltag: A robust and flexible visual fiducial system, Robotics and Automation (ICRA), 2011 IEEE International Conference on 2011, pp. 3400–3407.
[23] J. DeGol, T. Bretl, D. Hoiem, Chromatag: A colored marker and fast detection algorithm, 2017 1481–1490.
[24] F.J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Fractal markers: A new approach for long-range marker pose estimation under occlusion, IEEE Access 7 (2019) 169908–169919.
[25] R. Muñoz-Salinas, M.J. Marín-Jimenez, E. Yeguas-Bolivar, R. Medina-Carnicer, Mapping and localization from planar markers, Pattern Recogn. 73 (2018) 158–171.
[26] J. DeGol, T. Bretl, D. Hoiem, Improved Structure from Motion Using Fiducial Marker Matching: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III, 2018 281–296.
[27] D.S. Bolme, J.R. Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2010, pp. 2544–2550.
[28] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[29] M. Danelljan, F.S. Khan, M. Felsberg, J. v. d. Weijer, Adaptive color attributes for real-time visual tracking, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014 1090–1097.
[30] Y. Li, J. Zhu, A. Scale Adaptive, Kernel Correlation Filter Tracker with Feature Integration, Computer Vision - ECCV Workshops 2014, pp. 254–265.
[31] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, M. Kristan, Discriminative correlation filter tracker with channel and spatial reliability, Int. J. Comput. Vis. 126 (7) (2018) 671–688.
[32] C. Ma, J. Huang, X. Yang, M. Yang, Hierarchical convolutional features for visual tracking, 2015 IEEE International Conference on Computer Vision (ICCV) 2015, pp. 3074–3082.
[33] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking, 2015 IEEE International Conference on Computer Vision Workshop (ICCVW) 2015, pp. 621–629.
[34] M. Danelljan, A. Robinson, F. Khan, M. Felsberg, Beyond correlation filters: Learning continuous convolution operators for visual tracking, Springer International Publishing, 2016 472–488.
[35] M. Mueller, N. Smith, B. Ghanem, Context-aware correlation filter tracking, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, pp. 1387–1395.
[36] M. Danelljan, G. Häger, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking, 2015 IEEE International Conference on Computer Vision (ICCV) 2015, pp. 4310–4318.
[37] M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, Proceedings of the British Machine Vision Conference, BMVA Press 2014, pp. 1–11.
[38] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man, and Cybernetics 9 (1) (1979) 62–66.
[39] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision in C++ with the OpenCV Library, 2nd edition O'Reilly Media, Inc., 2013
[40] F.J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers, Image Vis. Comput. 76 (2018) 38–47.
[41] M. Fiala, Designing highly reliable fiducial markers, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2010) 1317–1324.
[42] K. Madsen, H.B. Nielsen, O. Tingleff, Methods for non-linear least squares problems (2nd Ed.), 2004.
[43] Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing 30 (1), 1985 32–46.
[44] 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.
[45] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, Computer Vision – ECCV 2012, Springer Berlin Heidelberg 2012, pp. 702–715.
[46] B. Babenko, M.-H. Yang, S. Belongie, Visual tracking with online multiple instance learning, 2009 IEEE Conference on Computer Vision and Pattern Recognition 2009, pp. 983–990.
[47] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1409–1422.
[48] Z. Kalal, K. Mikolajczyk, J. Matas, Forward-backward error: Automatic detection of tracking failures, 2010 20th International Conference on Pattern Recognition 2010, pp. 2756–2759.
[49] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, 1, 2006 47–56.
[50] J. Engel, T. Schöps, D. Cremers, LSD-SLAM: Large-scale direct monocular SLAM, European Conference on Compututer Visision (ECCV), 2014.

Chapter 5

Conclusions

The contributions presented in this thesis has solved some of the problems inherent to marker detection, allowing its use even under challenging conditions. On the other hand, strategies have been followed to maximize the efficiency of the marker detection system, allowing to obtain accurate pose estimations. The conclusions are summarized in the following points:

- We describe a new methodology for artificial marker detection, maximizing computational time while maintaining accuracy and robustness. Our method has been compared to state-of-art methods and is up to 40 times faster, achieving 1000 fps in 4K images processing without parallelization.
- This work proposes a multi-scale image representation strategy, as well as methods to determine the optimal resolution for detection, identification and corner estimation.
- A new type of marker is proposed, which can be detected over a greater range of distances than a traditional marker. The new marker is flexible and configurable allowing it to be used in a wide range of applications mainly oriented to robotics and augmented reality.
- A new RANSAC-based method for marker detection under occlusion. The keypoint-based algorithm uses the internal corners of the marker and its categories to validate the detection.

- A new approach for tracking of markers using a set of Discriminative Correlation Filters, following a multiscale strategy. It combines a proper scale selection and a corner upsampling strategy. Achieving HD image processing rates more than 200fps without parallelization.

- A new predictive marker detection strategy based on Discriminative Correlation Filters. This method has been compared with state-of-the-art SLAM methods obtaining higher accuracy, sensitivity and speed.

All the proposed methods are publicly available for research use, and integrated into the ArUco library[1], as well as all the datasets used in the experiments to facilitate their reproduction.

Finally, let me indicate that the first contribution to this thesis "Speeded up detection of squared fiducial markers" [32] has received at the moment 178 citations (Scopus source) and 300 citations (Google Scholar source). In addition, it is listed in the 1st position in the section " The most cited articles published since 2018" in the Image and Vision Computing Journal (Q1) and it has receiving in 2020 the "Editors Choice Award 2020".

[1] https://www.uco.es/investiga/grupos/ava/node/26

References

[1] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin, "Vision-based and marker-less surgical tool detection and tracking: a review of the literature," *Medical Image Analysis*, vol. 35, pp. 633–654, 2017.

[2] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[3] D. J. Mirota, M. Ishii, and G. D. Hager, "Vision-based navigation in image-guided interventions," *Annual review of biomedical engineering, 13*, p. 297–319, 2011.

[4] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A vision system for landing an unmanned aerial vehicle," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, vol. 2, pp. 1720–1727, Ieee, 2001.

[5] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision-based autonomous landing of an unmanned aerial vehicle," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 3, pp. 2799–2804, 2002.

[6] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1, pp. 52–73, 2007.

[7] M. Hunke and A. Waibel, "Face locating and tracking for human-computer interaction," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1277–1281, IEEE, 1994.

[8] D. Van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *International journal of virtual reality*, vol. 9, no. 2, pp. 1–20, 2010.

[9] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.

[10] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Compututer Visision (ECCV)*, September 2014.

[11] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[12] S.-h. Zhong, Y. Liu, and Q.-c. Chen, "Visual orientation inhomogeneity based scale-invariant feature transform," *Expert Syst. Appl.*, vol. 42, pp. 5658–5667, Aug. 2015.

[13] 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 Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[14] H. Kato and 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*, pp. 85–94, 1999.

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

[16] Q. Bonnard, S. Lemaignan, G. Zufferey, A. Mazzei, S. Cuendet, N. Li, A. Özgür, and P. Dillenbourg, "Chilitags 2: Robust fiducial markers for augmented reality and robotics.," 2013. http://chili.epfl.ch/software.

[17] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1317–1324, 2010.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[19] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2nd ed., 2013.

[20] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[21] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[22] X. Cortés and F. Serratosa, "Cooperative pose estimation of a fleet of robots based on interactive points alignment," *Expert Systems with Applications*, vol. 45, pp. 150 – 160, 2016.

[23] F. Ababsa and 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, pp. 431–435, 2004.

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

[25] K. Dorfmüller and H. Wirth, "Real-time hand and head tracking for virtual environments using infrared beacons," in *Proceedings of the International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, CAPTECH '98, (London, UK), pp. 113–127, Springer-Verlag, 1998.

[26] M. Ribo, A. Pinz, and A. L. Fuhrmann, "A new optical tracking system for virtual and augmented reality applications," in *In Proceedings of the IEEE Instrumentation and Measurement Technical Conference*, pp. 1932–1936, 2001.

[27] V. A. Knyaz and R. V. Sibiryakov, "The development of new coded targets for automated point identification and non-contact surface measurements," in *3D Surface Measurements, International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, part 5*, pp. 80–85, 1998.

[28] L. Naimark and E. Foxlin, "Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker," in *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, (Washington, DC, USA), pp. 27–, IEEE Computer Society, 2002.

[29] J. Rekimoto and Y. Ayatsuka, "Cybercode: designing augmented reality environments with visual tags," in *Proceedings of DARE 2000 on Designing augmented reality environments*, DARE '00, (New York, NY, USA), pp. 1–10, ACM, 2000.

[30] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1898–1903, 2019.

[31] H. Wang, X. Wang, G. Lu, and Y. Zhong, "Harco: Hierarchical fiducial markers for pose estimation in helicopter landing tasks," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1968–1973, 2015.

[32] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Speeded up detection of squared fiducial markers," *Image and Vision Computing*, vol. 76, pp. 38–47, 06 2018.

[33] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Fractal markers: A new approach for long-range marker pose estimation under occlusion," *IEEE Access*, vol. 7, pp. 169908–169919, 2019.

[34] F. J. Romero-Ramirez, R. Muñoz-Salinas, and R. Medina-Carnicer, "Tracking fiducial markers with discriminative correlation filters," *Image and Vision Computing*, vol. 107, p. 104094, 2021.

Appendix A

Impact factor report

- **Speeded up detection of squared fiducial markers.** Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, Rafael Medina-Carnicer. Image and Vision Computing, Volume 76, Pages 38-47, 2018.
  - Journal Impact Factor (2018): 2.733 (Q1)
  - Journal Rank (2018): 23/107
  - Category: Computer Science, Software Engineering
  - Source: Journal Citation Reports (JCR)
  - Cites number (Scopus): 178
  - Most cited article in Image and Vision Computing Journal since 2018, extracted from Scopus.
- **Fractal Markers: a new approach for long-range marker pose estimation under occlusion.** Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, Rafael Medina-Carnicer. IEEE Access, Volume 7, Pages 169908-169919, 2019.
  - Journal Impact Factor (2019): 3.745 (Q1)
  - Journal Rank (2019): 35/156
  - Category: Computer Science, Information Systems

  - Source: Journal Citation Reports (JCR)
  - Cites number (Scopus): 3

- **Tracking Fiducial Markers with Discriminative Correlation Filters.** Francisco J. Romero-Ramirez, Rafael Muñoz-Salinas, Rafael Medina-Carnicer. Image and Vision Computing, Volume 107, Pages 104094, 2021.
  - Journal Impact Factor (2019): 3.103 (Q1)
  - Journal Rank (2019): 21/108
  - Category: Computer Science, Software Engineering
  - Source: Journal Citation Reports (JCR)