A Fiducial Tag Invariant to Rotation, Translation, and Perspective Transformations

Heriberto Cruz Hernández, Luis Gerardo de la Fraga

PII: S0031-3203(18)30114-6
DOI: 10.1016/j.patcog.2018.03.024
Reference: PR 6502

To appear in: *Pattern Recognition*

Received date: 4 October 2016
Revised date: 23 March 2018
Accepted date: 25 March 2018

Highlights

- We test experimentally the use of the Computational Geometry con- struction called order type for auto-identification of a fiducial tag.
- We describe a procedure for detecting the fiducial tag from an image and the algorithms for computing an identification number from them.
- We do extensively tests to show that the proposed construction is invariant to translation, rotation, and perspective transformations.
- We analyze which is the maximum noise in point positions for all points within a set that allows to compute correctly the order type in that set of points.
- As the proposed new tags are invariant to perspective transformation, it is not necessary to remove in the tag image this distortion in order to recover its associated identification number.
- With a tag of 21×21 cm and images of size 640×480 pixels, our results show that the tag and its associated ID can be recovered if the tag is tilt less than 81°, including perspective distortion, and at a distance less than 62 cm. Also, up to 3,472 of such IDs are available.

A Fiducial Tag Invariant to Rotation, Translation, and Perspective Transformations

Heriberto Cruz Hernández, Luis Gerardo de la Fraga*
CINVESTAV, Computer Science Department. Av. IPN 2508, 07360 Mexico City, Mexico

Abstract

This work introduces a novel visual fiducial tag appropriate for applications of automatic identification. The proposed tag is based in Order Type, a construction defined in Computational Geometry, which is invariant to 3D translation, rotation, and projective transformations. Three main contributions are presented: first we describe the design of the proposed tags, the procedures for detecting them from an image, and the algorithms for computing an identifier from them. Second, we analyze the feasibility of the proposal in three different conditions of tag's rotation, distance to the tag, and the effect of noise in point positions for the recognition process. Third, we show the applicability of the proposed tags with simulated images. The conducted experiments indicate that the tags are very robust to the image generation process, suitable for automatic identification up to 3 472 different tags, and also for the pose estimation in Computer Vision applications.

Keywords: Order Type, Computational Geometry, visual fiducial tag, point pattern matching, automatic identification, Computer Vision

1. Introduction

Visual fiducial tags, or fiducials, are artificial landmarks that are easy to detect and to identify from images taken by digital cameras. They are high

*Corresponding author
Email addresses: `hcruz@computacion.cs.cinvestav.mx` (Heriberto Cruz Hernández), `fraga@cs.cinvestav.mx` (Luis Gerardo de la Fraga)

reflectance bidimensional patterns with known structure that encode a unique ID [1].

Fiducial tags have an special importance in the Computer Vision and Augmented Reality [2, 3, 4, 5, 6] fields because they are an easy approach for helping a computer system to identify objects and their location in an scene seen through a camera. Some applications of them are object identification, tracking, camera calibration, and robot localization [3, 6, 7, 8], and very recently for robot localization and mapping [9].

In recent decades, the use of visual fiducial tags has become very popular for Augmented Reality and robotics applications since they allow to perform automatic identification, and also because they are useful to estimate the relative position and orientation between the camera and the tag. They are considered the cheapest technology since can be generated with conventional printers and can be detected with low cost cameras as those embedded in personal computers or mobile cell phones [10].

Unlike bidimensional barcodes like QR [11] or Datamatrix codes [10] that have the capability to encode big amounts of information (text or numbers), fiducials are provided with an smaller set of previously known numbers, or identificators (IDs), called a dictionary. The reason for this difference comes from the fact that bidimensional barcodes are assumed to be read one per image and also to occupy the totality of the image field, by contrast, fiducial tags are intended to be read from larger distances, in conditions of 3D rotation and multiple instances of them can be read from a single image [12].

For most of the existing visual fiducials tags in literature, three essential steps are performed for their decodification from an image: Tag identification, Data area correction, and Data area decodification.

1) Tag identification consists in finding those segments of the image that seem to contain a tag. This identification is done by finding the constant parts of the tag like quad borders or squares at the corners, e.g., three of the later for QR codes.

2) Since the pattern in the image can be distorted by a projection or it can be

occluded, the data area correction step consists in getting a corrected version of the image. This step is commonly solved by computing a homography between the tag in the image and the known structure of the tag to correct the image [12, 13, 14]. The homography is a 3×3 matrix (i.e. a projective transformation) with 8 DOF that is used to estimate the position in the image of specific elements in the structure. This step is crucial in most of the existing approaches and the correct tag decodification depends on the correctness of this step.

3) Once data area is corrected, information is decoded. For those approaches based in bits, this step consists in computing the encoded ID by reading all bits from the image, applying their verification, and correction.

Since most of the existing visual fiducial tags perform the homography estimation for the identification step, in this paper we propose new visual fiducial tags, based in the combinatorial invariant Order Type, that are invariant to rotation, translation and perspective transformations.

This article is organized as follows: in Sec. 2 we present a review of the state of the art visual fiducial tags, in Sec. 3 we describe Order Type, its definition and all the related concepts used in this paper. In Sec. 4 we describe the design for the proposed markers. In Sec. 5 we present all experiments performed to validate our proposal. In Sec. 6 a brief discussion about this work is given, and finally, in Sec. 7, conclusions and future work are drawn.

2. Related work

The way the ID is encoded inside the pattern is one of the main aspects to classify the existing works. The first and more common approach are the bidimensional binary patterns. In this category we find AprilTags [13], ARTags [15], BinARyID [16] (See Fig. 1), and ArUco [17]. They all are composed by a black squared border with the binary pattern inside. The black squared border is mainly the first object in the scene to detect because its four corners are used for estimating the homography transformation for the data area correction step. Each work in this category apply different strategies to try to maximize the inter-tag distance. Inter-tag distance is the way to measure how different are valid tags (for binary approaches this is measured in terms of the Hamming distance), a higher inter-tag distance a lower inter-tag confusion.

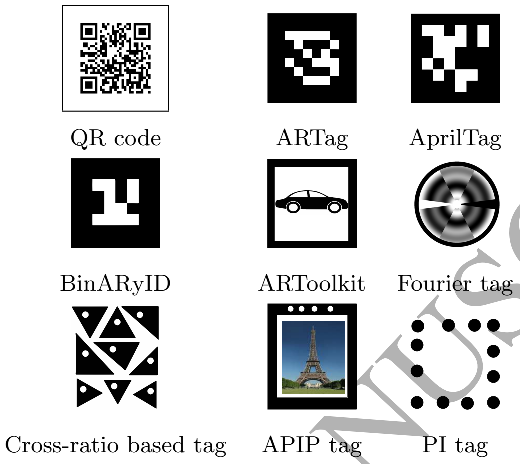


Figure 1: Several proposed visual fiducial tags instances.

AprilTags are composed by a white background along a black squared border with a 6×6 binary pattern inside. Authors in [13] choose valid codes using the Hamming distance in such way that all valid codes have a distance of at least 10 bits from each other. As complementary strategy, AprilTags define a heuristic for selecting as valid codes just them that are enough complex. Complexity is defined as the number of rectangles conforming the binary pattern. The Hamming distance along the number of rectangles are tuning variables that allows the user to choose as a trade off the number of valid tags against the inter-tag confusion.

ARTags are also composed by a squared border with a 6×6 binary pattern inside. From the possible 36 bits only 10 bits are used for ID encoding. The remaining bits are used for providing robustness, through digital information techniques as *Cyclical Redundancy Check* (CRC) along *forward error correction* (FEC). These algorithms allow to identify when a tag is correctly read and also to correct errors respectively.

An important aspect to mention about ARTags and AprilTags is that they increase robustness at cost of a reduced number of valid tags. ARTags count with 2002 different codes, 1001 with white background and black border and 1001 for the inverse case. In the AprilTags case it is possible to reduce or increase the number of different valid codes but in [12] they report 2221 different valid codes considering a 10 bits for Hamming distance and at least 10 rectangles for complexity.

A similar approach to April tags are the ArUco markers proposed in [17], its authors propose a configurable dictionary (in terms of bits and number of markers) and an stochastic algorithm to construct it. The dictionary can also be generated by implementing Mixed Integer Linear programming to obtain optimal marker dictionaries in terms of inter-marker distance as proposes in [18].

Other and less common approaches are those that do not use binary patterns. In this category we find ARToolkit markers [19], Fourier Tags, the cross ratio based tags proposed in [20], APIP (algorithm for projective invariant patterns) tags [21], and PI (projective invariant) tags [22] (see Fig. 1).

Fourier tags [23] are circular patterns with black border that encode binary data into gray scale radially symmetric structure. They exploit the fact that content in low frequencies is robust to many image degradations to avoid the abrupt degradation issue, i.e, the effect when an already fully detected tag suddenly is undetected by the system. This kind of tags can be affected by the presence of blurring, the quantification error of the printer and the correct decodification depends upon the accurate boundary detection, i.e., the black circle seen as an ellipse in images.

ARToolkit markers [19] (See Fig. 1) are squared patterns composed by a black squared margin with any image inside. They are based in pattern classification, specifically in image correlation and one of the main aspects to stand out for template markers is that the system must be trained offline with the accepted images.

The authors in [20] propose a tag composed by black triangles, each with a white circle inside. They use the cross-ratio projective invariant as identification mechanism and they propose to use the triangle vertices as additional points for pose (location and orientation) estimation. Authors do not define the number of possible tags, it is not clear how to assign a single ID to a tag, and cross-ratio is very sensible to the errors in the vertices position estimation.

In [21], Teixeira et al. propose APIP, a tag conformed by four collinear circles. In this work, the ID codification is based in the cross ratio of all the collinear points. Points are detected as the centroid of the ellipses detected in image space. The method requires a training step and is only suitable for automatic identification. The pose estimation is performed using the black squared border. Using the centroid of ellipses as the image feature increase the noise in these APIP tags, because the center of a circle is not equal to the centroid of the projected circle (an ellipse) [24].

In [22], Bergamasco et al. propose PI tags. These tags are also based in collinearity of points, cross ratio, angular ordering, and invariance of the class of ellipses (for projective invariants). PI tags work for automatic identification and also for pose estimation. They are mainly based in the cross ratio of collinear ellipses centroids, and they introduce redundancy of patterns to allow occlusion resistance. Since PI-tags recognition step is mainly based in cross ratio, the number of different tags is limited due to the cross ratio noise sensitiveness. To reduce the error introduced by using ellipse centroids (which are not invariant to projective transformations) authors propose to use small circles, which difficult their detection.

Most tags that allow pose estimation need to solve the point matching problem. At least four points are needed to estimate a homography between a tag's squared border and the corresponding tag seen in the image that allows its relative orientation and translation estimation. The matching problem has been widely studied for general images [25, 26, 27] with many points (dozens or hundreds), and for 3D point clouds [28], but for the visual markers, most of approaches with squared border solve the matching using only the four corners. In this case the ambiguity introduced by the symmetry at 0, 90, 180, and 270 degrees is solved by additional mechanisms after the homography is estimated. The proposed Order Type tags solve the point matching at the same time when the marker is identified, do not present the symmetry ambiguity, and the identification is made with more than four points, which significantly differs from other approaches.

In Tab. 1 we show a comparison among several fiducials approaches.

Table 1: Comparison among existing six tag approaches an our proposed order type tag

Approach	AprilTags	ARTags	ARToolkit	APIP	Pi-tags	Order Type Tags
Projective invariant	No	No	No	Yes	Yes	Yes
Image feature	B/W squares	B/W squares	Whole image	Ellipses centroids	Ellipses centroids	Triangles vertices
Known number of different tags	Up to 2002	Up to 1001	Unknown	Unknown	Unknown	Up to 3 472
Require image rectification	Yes	Yes	Yes	No	No	No
Occlusion robustness	Yes	Yes	Yes	Possible	Yes	Possible
Allow pose estimation	Yes	Yes	Yes	No	Yes	Yes
Recognition method	Hamming distance	Hamming distance	Correlation	Cross ratio difference	Cross ratio difference	Order type
Decodification complexity	N/A	N/A	N/A	N/A	$O(n^4)$	$O(n^3)$
Require training	No	No	Yes	Yes	Yes	No

3. Order types

In the Computational Geometry field, Goodman and Pollack [29] first introduced *order type* (OT) as a method to describe point sets in terms of the orientation of subsets of three points. OT can be understood as a conceptual way for describing point sets in the space and it is considered one of the most fundamental combinatorial descriptions of points on the plane. It encodes for each triplet of points its orientation and thus reflects most of the combinatorial properties of a given set, avoiding the use of metric information.

The OT of a point set C is a function that assigns to each triple of points in C its orientation. We say that two sets of points C_1 and C_2 are equivalent if they have the same OT.

OT is stored using an Order Type Representation (OTR). OTRs can be seen as data structures that quantify the triplets orientations. Many of them have been proposed [30] but one of the most compact is the λ-matrix.

λ-matrix is an OTR originally proposed by Goodman and Pollack [29]. It is a $n \times n$ matrix whose each entry $\lambda(i, j)$ represents the number of points in the set that are on the left (positive) side of the oriented line through $\boldsymbol{p}_i, \boldsymbol{p}_j$, for $i \neq j$ (see Fig. 2).

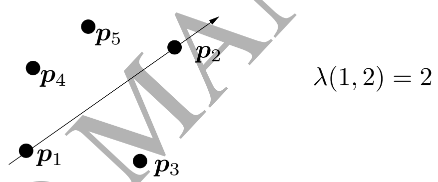


Figure 2: A λ-matrix entry. Considering a directed line that passes from $\boldsymbol{p}_1$ to $\boldsymbol{p}_2$, the λ-matrix entry value is the number of points that are in left side the line, i.e., the number of points $\boldsymbol{p}$ that satisfy the condition $A(\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}) > 0$, where A computes the signed area of the triangle $\boldsymbol{p}_i, \boldsymbol{p}_j, \boldsymbol{p}$.

An important aspect to mention is that λ-matrix depends on points labeling. Two different labelings of the same point set will correspond to two different λ-matrices. Although λ-matrix is sensible to point set labeling, the OT is not [29].

If two point sets C_1 and C_2 have the same λ-matrix and thus the same OT, C_1 and C_2 will be combinatorially equivalent. Since there are $n!$ possible labelings in a point set, for each point set there will be $n!$ associated λ-matrices. A naive method to determine if two unlabeled point sets are combinatorially equivalent is to fix a labeling in C_1, compute the associated λ-matrix and compute the $n!$

λ-matrices of C_2 until finding a coincidence of matrices.

A more efficient method is based in canonical order [29]. Canonical order is a way to label elements in a point set in a counterclockwise way starting by those points on the convex hull. A pseudocode to compute all canonical orders is shown in Algorithm 1 and one example is illustrated in Fig. 3.

Algorithm 1 All canonical orderings computation

Require: A point set C
Ensure: All canonical orderings of C

$L \leftarrow$ Compute convex hull conv(C) of C.
$L_t \leftarrow L + (C - L)$ ▷ L_t is a list with all k points
for each point $\boldsymbol{p}$ in L **do**
    swap($\boldsymbol{p}_1$, $\boldsymbol{p}$)
    $L_c \leftarrow$ sort L_t for points $2, 3, \ldots, k$, using comparison() procedure.
    ▷ Points in L_t are sorted counterclockwise with respect to $\boldsymbol{p}_1$.
    ▷ L_c form a canonical ordering.
end for

procedure COMPARISON($\boldsymbol{p}_i$, $\boldsymbol{p}_j$)
    $v = A(\boldsymbol{p}_1, \boldsymbol{p}_i, \boldsymbol{p}_j)$ ▷ A calculates signed area
    if ($v > 0$) **then return** -1
    else if ($v < 0$) **then return** 1
    else ▷ v is equal to 0
        if $d(\boldsymbol{p}_1, \boldsymbol{p}_i) < d(\boldsymbol{p}_1, \boldsymbol{p}_i)$ **then**
            return -1 ▷ d calculates Euclidean distance
        else return 1
        end if
    end if
end procedure

Since we can choose $m = |\text{conv}(C)|$ different initial points, there will also be m canonical orderings. The Graham's algorithm [31] for convex hull computation can be used to compute the canonical orderings. The algorithm sorts points in counterclockwise order and later discards all points that do not belong to the convex hull. The first part of Graham's algorithm, the points ordering, is shown within Algorithm 1.

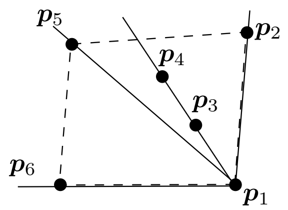


Figure 3: One of the four canonical orderings for $C = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_6\}$. Points are ordered counterclockwise with respect to $\boldsymbol{p}_1$.

.

The method in [29] proposes to compute the λ-matrix associated to one canonical order of C_1, and test the λ-matrices of all canonical orderings of C_2. With this method the total number of λ-matrices comparisons is at most n in the case when all points of C_2 are on $\mathrm{conv}(C_2)$.

Considering that:

- A high number of order types exists for small point sets cardinality.
- There not exists any visual fiducial tag based in combinatorial properties.
- In most of the existing approaches the image correction (by homography estimation or equivalent) is a crucial step.

Then we propose to use the OT concept as the base to design Auto-ID tags that avoid the homography estimation, therefore they are invariant to projective distortion, and also that allows us to obtain a competitive different number of tags.

4. Visual tags based in Order Types

The number of OTs that exist on the plane is finite and it depends on the cardinality of the point set. The possible OTs that can be generated up to 10 points have been studied by Oswin Aichholzer et al. [32], and these authors provide a database with the enumeration of the possible OTs that can be generated. The provided database contains a point set instance for each of the enumerated OTs. A summary of the OTs that conform the database is shown in Tab. 2. Since the database contains an instance for each of the existing order types, we used this database as reference for constructing the proposed Order Type Tags (OTTs) and for our experiments. We used only point sets with cardinality less or equal to eight. This choice is because the points on those point sets are given as pairs of 8 bits integers that means they require a resolution of 256×256 pixels that makes them printable by conventional printers and also because the resolution of the common hand held cameras is enough to read them.

For simplicity, we denote as C^k the set with point subsets of the same cardinality, where $k \in \{3, 4, \ldots, 9, 10\}$ is the cardinality, and each subset instance in C^k as C^k_l, where l is the number of instance.

Table 2: Oswin Aichholzer et al. order type database summary.

Set	$\lvert C^k \rvert$ = Number of OTs
C^3	1
C^4	2
C^5	3
C^6	16
C^7	135
C^8	3315
C^9	158 817
C^{10}	14 309 547
C^{11}	2 334 512 907

4.1. Order Type Tags

In this section we describe the proposed tags. We first describe the proposed structure for OTTs, i.e. their elements and organization, and then we describe the process and sub-processes involved up to ID computation.

Order Type Tags (OTTs) are composed by three main elements: quiet area, data area, and the tag points, an instance of them is shown in Fig. 4. Quiet area is a white area that contains all other elements of the tag, its purpose is to help the data area to be detected complete since it serves as separator for data area and other objects in the scene seen in the image. Data area is a black square that simultaneously works as finding pattern. Its purpose is to serve as the most easy to identify object (it is supposed to be the biggest black object on the scene) at the same time it serves to delimit the image segment where tag points are contained. Tag points constitute the point set C and their arrangement define the OT and the ID. They are given as triangles vertices to help the system to find them in an image. Triangles allow to compute their vertices positions at sub-pixel precision.

For defining triangles in OTTs we take a point set from database and we manually define triangles by looking to use the least number of triangles but assuring that all points are used as vertex of at least one triangle.

Order Type Tags decoding consists in three phases: potential OTTs detection, point set estimation, and ID computation. These three phases will be explained in the subsequent paragraphs.

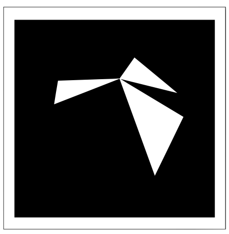


Figure 4: Instance of the proposed Order Type Tags. The tag was constructed using the 3rd point set with cardinality 7 of the database in [32]. Point set $C = \{[206, 159], [214, 127], [176, 49], [42, 144], [47, 175], [129, 178], [149, 206]\}$.

4.1.1. Potential Order Type Tags detection

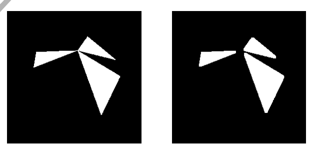
This task consists in identifying from an image those segments that could potentially contain a valid OTT. We first adequate image to make the potential Order Type Tags (POTTs) easy to detect in the image. We convert images in color to grayscale [33] and we apply Otsu's thresholding method for binarization. In the binary image we look for the groups of black connected pixels.

To reduce the false positive rate that can be caused by small pixel groups or noise we only consider those objects that area is greater than a threshold. Not all objects detected in this phase correspond to the data area of an OTT but the point set estimation phase allows to reject those that does not have the OTT structure.

4.1.2. Point set estimation

The purpose of this phase is to find all points that conform the point set. We do this by detecting all triangles in the OTT with the objective to estimate each triangle vertex. We treat each POTT as a separate binary image. For each POTT we apply the following procedure:

We apply the opening morphological operator [34] to the white pixels with the objective of removing small groups of white pixels coming from noise. This step also separates those triangles that share vertices (See Fig. 5).



a) b)

Figure 5: Opening on a potential Order Type Tag. a) POTT before opening operator (Join triangles). b) POTT after opening operator (Separated triangles).

We consider as triangles those white pixel objects with the highest area inside the POTT. For each detected triangle is necessary to estimate its vertices. We do this task by selecting as vertices the two farthest pixels in the object. Then, for the third vertex, we look for the point with maximum distance from the line formed by the two already found vertices. Using this first vertices approximation we segment the perimeter pixels in in three chains, each chain corresponding to the pixels on a triangle's side (See Fig. 6). For each pixels chain we apply linear regression using Principal Component Analysis algorithm and triangle vertices are computed as the intersection of two lines as shown in Fig. 7.

Shared vertices can result in different estimations, one for each triangle. To solve this problem, the estimated vertices are clustered. We fix the number of clusters as the number of points of the tag and we consider as final vertices the centroids of the formed clusters.

In this phase if the number of estimated pixels differs to the point set cardinality, POTT is rejected and ID computation cannot be performed.

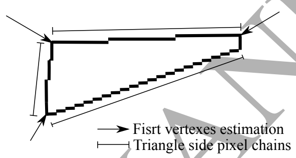


Figure 6: Triangle vertices first estimation and side pixel chains.

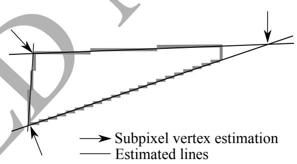


Figure 7: Estimated lines and subpixel vertices estimation.

4.1.3. ID computation

We define as ID a single and unique number that identify an OTT. The objective of this phase is that for a same OTT we want to be able to obtain the same ID even if it is affected by projective and rigid transformations, pixel conversion, and positional noise. As ID we propose to use the λ-matrix corresponding to the minimal lexicographical of all the canonical orderings of a point set. The procedure to compute the OTT ID is described in Algorithm 2, and one example of the output data with one instance of C^5 is shown in Fig. 8.

Algorithm 2 Order Type Tag ID computation

Require: A point set C

Ensure: Point set ID

1: Compute all canonical orderings as described in Algorithm 1.

2: For each canonical ordering compute the associated λ-matrix.

3: Order lexicographically all λ-matrices obtained in step 2, and choose as ID the minimal lexicographical.

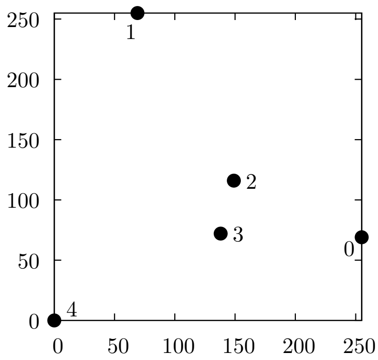
The ID computation in Algorithm 2 is invariant to translation and rotation of the point set C.

Observing matrix at the bottom in Fig. 8, its last row is $[3, 0, 1, 2, -]$; this row corresponds to point 4. The values of this row are calculated as follows: if it is traced a line from point 4 to 0 (see the last set of point in Fig. 8), the number of points to the left of this line is 3 (it is the first element value in the row). If the line is traced from point 4 to 1, there is not any point to the left of this line, thus the second element in the row is 0, and so on.

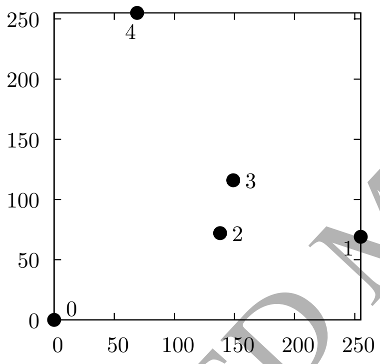
4.1.4. Homography estimation

Homography estimation requires to solve the matching problem, i.e., to find the correct correspondence between model and image points, in our case marker vertices. A correct matching allows to compute a correct homography, which can be used to perform camera calibration and pose estimation. In our approach we can solve the matching problem through an invariant labeling. Two point sets (one from database and other from image in the tag decodification step) are matched by the labeling corresponding to the minimal lexicographical $\lambda-$matrix from the canonical orderings, i.e., points with the same label ID are correspondences (see Fig. 8).

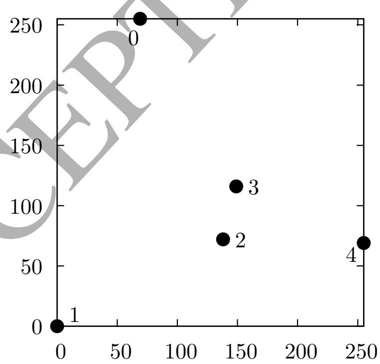
Although OT allows point matching, not all OTs are suitable for this task, because some OTs have more than one minimal lexicographical matrix. Let E to be the set of points with a single lexicographical matrix, $E_k = \{C_k^l | C_k^l$ has only one minimal lexicographical matrix $\}$. In Tab. 3 we show the number of



$$\begin{bmatrix} - & 3 & 2 & 1 & 0 \\ 0 & - & 1 & 2 & 3 \\ 1 & 2 & - & 1 & 2 \\ 2 & 1 & 2 & - & 1 \\ 3 & 0 & 1 & 2 & - \end{bmatrix}$$



$$\begin{bmatrix} - & 3 & 2 & 1 & 0 \\ 0 & - & 1 & 2 & 3 \\ 1 & 2 & - & 2 & 1 \\ 2 & 1 & 1 & - & 2 \\ 3 & 0 & 2 & 1 & - \end{bmatrix}$$



$$\begin{bmatrix} - & 3 & 2 & 1 & 0 \\ 0 & - & 2 & 1 & 3 \\ 1 & 1 & - & 2 & 2 \\ 2 & 2 & 1 & - & 1 \\ 3 & 0 & 1 & 2 & - \end{bmatrix}$$

Figure 8: Example of the three canonical orderings and their associated λ-matrices of points set $\{[0,0],\ [69,255],\ [149,116],\ [138,72],\ [255,69]\} \in C^5$. The λ-matrices are sorted top to bottom according to their lexicographical value. The ID associated to this set of points is the λ-matrix at the top.

OTs suitable for point matching from E^5 to E^8. The OT shown in Fig. 8 has only one minimal lexicographical matrix, thus it is suitable for point matching.

Table 3: The number of OTs suitable for point matching.

Set E^k	Number of OTs
E^5	2
E^6	11
E^7	131
E^8	3303

5. Experiments

To validate our proposal we performed various experiments to test different aspects about OT and its use in the proposed OTT. First, we analyzed how sensible is OT to perspective transformation in the image generation process, i.e., how feasible is to recover the OT from images transformed by 3D rotations, the projective transformation, and the pixel generation introduced by a camera. This experiment is described in detail in Sec. 5.1. A similar experiment but varying the distance from the camera to the tag is described in Sec. 5.2. The third experiment consisted in analyzing the sensibility of the proposed tag to the noise in the vertices position estimation. This last experiment is further detailed in Sec. 5.3.

5.1. Order type robustness to perspective and image generation

When acquiring an image of a scene with a camera, position of elements in the image change their position in a nonlinear way. This change is caused by the image generation process, i.e., the simultaneous effects of:

- The 3D rotations and translations of the camera (or the objects).
- Projective non linear transformation caused by the perspective introduced by the camera (related to the focal distance) and its obliquity.

- Pixel generation caused by the finite resolution of the camera (discretization errors).

The aim of this experiment was to analyze in an exhaustive approach how OT changes as function of rotation, tilt and perspective of the image generation. We used the sets C^7 and C^8 to generate synthetic scenes with a camera looking to a point subset. For each C_l^k we computed the ground truth associated ID using the Algorithm 2. Then, we generated synthetic images with C_l^k under different amounts of rotation and with the camera at fixed distance. From each synthetic image we computed ID_i to test how effective is to correctly recover the ground truth ID from images with tilt, rotation, perspective, and pixel generation.

For image generation we used the pinhole [35] camera model $\lambda \boldsymbol{p} = M\boldsymbol{P} = K[R|\boldsymbol{t}]\boldsymbol{P}$, where $\boldsymbol{p} = [u, v, 1]^T$ is a 2D point on an image in homogeneous coordinates (it represents a pixel in an image) and $\boldsymbol{P} = [x, y, z, 1]^T$ is a point in the 3D model (the scene). M is a 3×4 transformation matrix that holds the information about the camera and its position and orientation with respect to the scene. M is defined as $M = K[R|\boldsymbol{t}]$, where $K \in \mathbb{R}^{3\times3}$ is a matrix that holds the intrinsic parameters: focal distance $\{f_x, f_y\}$, obliquity of camera axes o and the position of the principal point of image $\{u_0, v_0\}$ as shown in Eq. (1). $R = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$ is a rotation matrix expressed by three Euler angles $\{\theta_1, \theta_2, \theta_3\}$ and $\boldsymbol{t} = [t_1, t_2, t_3]^{\mathrm{T}}$ is a translation vector. The parameters held by R and $\boldsymbol{t}$ are known as extrinsic parameters. As pixel generation we truncated the transformed points $\boldsymbol{p}$ to the closest integer value, i.e., $\boldsymbol{p} = [\lfloor u + 0.5\rfloor, \lfloor v + 0.5\rfloor, 1]^T$, where $\lfloor\cdot\rfloor$ is the floor operator.

$$K = \begin{bmatrix} f_x & o & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1}$$

We propose the camera with the configuration shown in Tab. 4. We assume each point set lies on the plane $z = 0$. Each point set was translated to place its centroid in the origin of the coordinate system.

Table 4: Parameters used to test the robustness to perspective and image generation process.

Parameter	Value
Image resolution:	640×480 pixels
Focal distance:	$f_x = f_y = 1000$
Principal point:	Image center at $(u_0, v_0) = (320, 240)$
Obliquity:	$o = 0$
Distance to the tag	1000 a.u.

We generated 3 240 images for each point set C_l^k rotating the plane around z axis from $\theta_1 = 0°$ to $\theta_1 = 359°$ in steps of $10°$, and changing tilt by rotating a different axis from $\theta_2 = 0°$ (without tilt) to $\theta_2 = 89°$ (high tilt) in steps of $1°$. The rotation of the other axis, θ_3, is always fixed to a value of $0°$. The camera is in the position shown in Fig. 9.

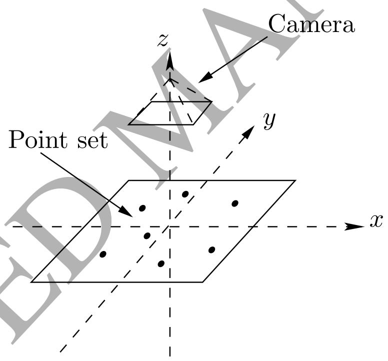


Figure 9: Camera plane and plane position on the 3D coordinate system.

The number of successful OT recoveries, the number of failed cases and the percent of success for our experiment are shown in Tab. 5. The histograms of failures according to the value of θ_2 angle are shown in Fig. 10, observing this figure, for seven points, 100% of failures are at $\theta_2 > 80°$; for eight points 99.94% of failures happen at $\theta_2 > 65°$.

Table 5: Results for simulations of OT recovering from images with rotation.

	7 points	8 points
Number of analyzed point sets	135	3 315
Images analyzed	437 400	10 740 600
Successful recovered IDs	427 437	10 093 612
Failed recovered IDs	9 963	646 988
Percent of success	97.72%	93.98%

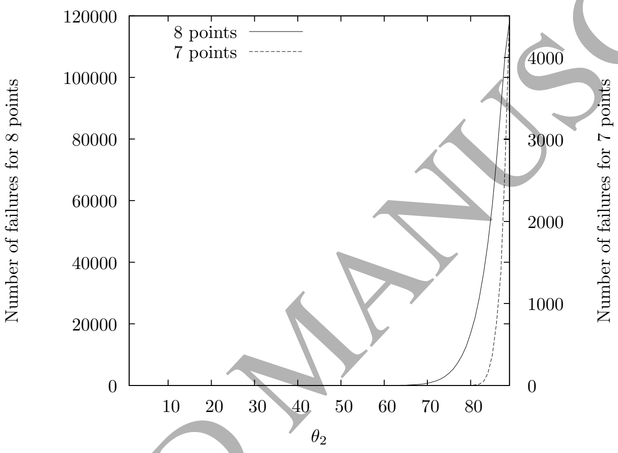


Figure 10: Histograms of failures of OT recoveries for rotation in θ_2.

5.2. Order type robustness to distance and image generation

As a complementary experiment to the described in Sec. 5.1 in this section we tested OT for distance variations. We maintain the same camera characteristics used in Sec. 5.1 but now we fixed θ_2 angle to zero, and the distance between camera and point set is varied from 500 to 10 000 in steps of 10 unities. The number of realized experiments and the summary of successful and failed cases is shown in Tab. 6. The histograms of failures according to the distance are shown in Fig. 11, observing this figure, for seven points, 100% of failures are at $d > 5580$; for eight points 99.95% of failures happened at $d > 2440$.

Table 6: Results for simulations of Order Type recovering varying distance.

	7 points	8 points
Number of analyzed point sets	135	3 315
Images analyzed	128 250	3 149 250
Successful recovered IDs	125 498	2 232 380
Failed recovered IDs	2 752	916 870
Percent of success	97.85%	70.89%

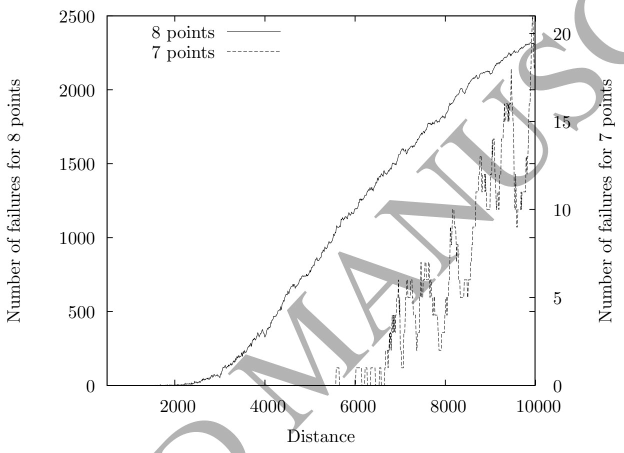


Figure 11: Histograms of failures of OT recoveries according to the distance to the tag.

5.3. Maximal Perturbation analysis

We are going here to analyze how noise in the position of the points affect the tag recognition. This noise affects the tag recognition in real conditions, and is generated from different sources as: camera resolution, camera lens distortion, camera movement, tag movement, tag bending, or triangle vertices position estimation.

The idea is to calculate which is the maximum movement of the points of a point set without change its OT. In Fig. 12 four points are drawn, a line pass through points $\boldsymbol{p}_1$ to $\boldsymbol{p}_3$; for this configuration $\lambda(1,3) = 1$ (one point is to the left to the line). The nearest distance from $\boldsymbol{p}_2$ to the line is equal to the radius of the bigger circle. It is possible to observe here that OT maintains unchanged if points are moved inside the smaller circles around them, with radius equal to a half of the orthogonal distance between point $\boldsymbol{p}_2$ and the line (a half of the radius of the bigger circle). We denote this distance as the Maximal Perturbation of C^k, or MP(C^k).

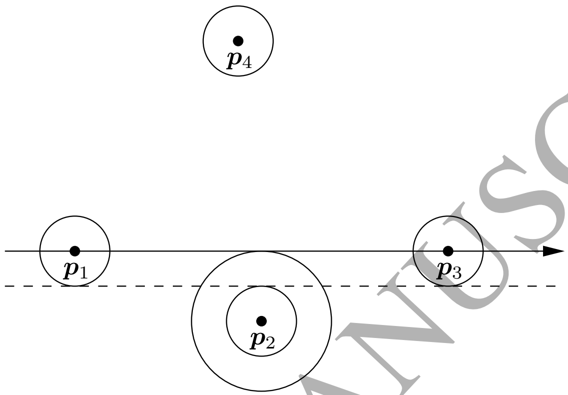


Figure 12: Four points on a plane, the bigger circle radius is the orthogonal distance from point $\boldsymbol{p}_2$ to the line through $\boldsymbol{p}_1$ and $\boldsymbol{p}_3$. $\lambda(1,3) = 1$ and points can be in any position inside the small circles without changing the OT. If both points $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, or points $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ cross simultaneously the dashed line, the OT changes (figure becomes a triangle with another point inside).

MP(C^k) is the maximal displacement that all the points in all instances C^k_l can have without changing its associated OT. Notice that for each C^k_l MP is distinct.

Allow us define the set $D^k = \{C^k_l | \mathrm{MP}(C^k_l) \leq v\}$, i.e., those instances C^k_l that have its associated MP(C^k_l) less than a given v value. In Tab. 7 we counted the cardinality of the D^k set, this value represents the number of distinct OTs, for $v = \{0.5, 1.0, \ldots, 9.0\}$.

As can be seen in Tab. 7, for $v = 6.0$ and eight points there is possible to get only 15 distinct OTs, i.e, if it is allowed noise in each point position of at most 6 a.u., then it is possible to get only 15 OTs. For five points it is possible to get 3 different OTs even with higher values of $v = 9.0$.

It is possible to increase the noise that affects the points position but this decreases the number of distinct OTs that can be obtained.

Table 7: Number of OTs for different noise allowed in point positions. Bold numbers indicate the maximum number of existing OTs for $k = \{8, 7, 6, 5\}$.

v	$\|D^8\|$	$\|D^7\|$	$\|D^6\|$	$\|D^5\|$
0.5	**3315**	**135**	**16**	**3**
1.0	3296	**135**	**16**	**3**
1.5	1240	**135**	**16**	**3**
2.0	642	**135**	**16**	**3**
2.5	371	**135**	**16**	**3**
3.0	231	86	**16**	**3**
3.5	135	60	**16**	**3**
4.0	83	47	**16**	**3**
4.5	56	32	**16**	**3**
5.0	37	26	**16**	**3**
5.5	26	18	**16**	**3**
6.0	15	15	**16**	**3**
6.5	10	8	**16**	**3**
7.0	6	7	12	**3**
7.5	5	4	10	**3**
8.0	4	3	8	**3**
8.5	3	3	6	**3**
9.0	3	3	5	**3**

5.4. Order Type Tags rotation test

For testing our proposal of the OTT design with we implemented the OTT decoder as described in Sec. 4. We arbitrarily used the tag shown in Fig. 4 to generate artificial scenes with the help of the ray-tracer program pov-ray [36]. For scene generation we considered the specifications in Tab. 8. We generated a total of 179 ray-traced images, some instances of them are shown in Fig. 13.

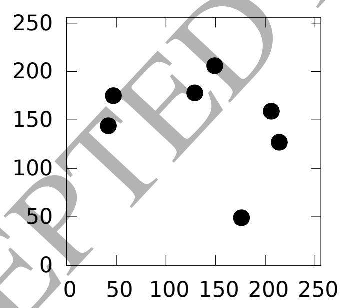
Table 8: Experiment conditions for Order Type Tag rotation experiment.

Algorithms parameters	
Area threshold	3% of the image width
Scene characteristics	
Tag size	21 × 21 cm
tag-camera distance	50 cm
Rotation angles	From $-89°$ to $89°$ in $1°$ steps
Camera characteristics	
Focal distance f	700 pixels
Obliquity o	0
Image size	640×480 pixels

As ground truth ID we computed the minimal canonical λ-matrix. We show the point set distribution along the ground truth ID in Tab. 9.

Table 9: Experiment conditions for Order Type Tag rotation experiment.

Point set distribution



Associated λ-matrix

$$\begin{bmatrix} - & 5 & 4 & 3 & 2 & 1 & 0 \\ 0 & - & 2 & 5 & 4 & 3 & 1 \\ 1 & 3 & - & 4 & 3 & 2 & 2 \\ 2 & 0 & 1 & - & 5 & 4 & 3 \\ 3 & 1 & 2 & 0 & - & 5 & 4 \\ 4 & 2 & 3 & 1 & 0 & - & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & - \end{bmatrix}$$

From the 179 images, 165 images were successfully detected and decoded. For the complementary 14 images our implementation did not detected the marker on the image thus the ID could not be decoded. These 14 images were those images with the highest rotation, seven images from $\theta_2 = [89°, 83°]$ and seven images for $\theta_2 = [-82°, -89°]$. The percent of success of detected correct IDs for this experiment was 92.17%.

ACCEPTED MANUSCRIPT

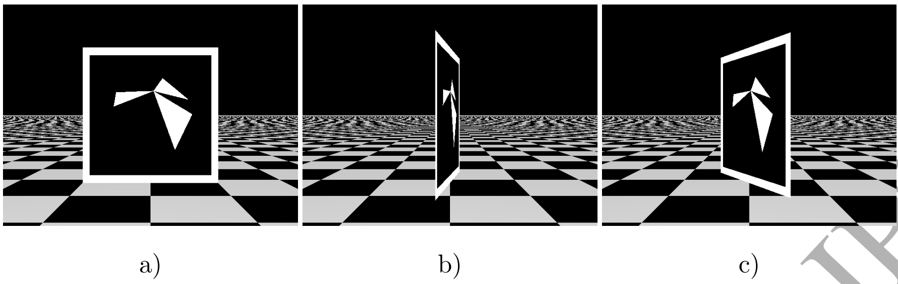


Figure 13: Instances of the ray-traced images for the rotation experiment. a) Tag with 0° of rotation. b) Tag with 80° of rotation. c) Tag with −60° of rotation..

5.5. Order Type Tags distance test

For this experiment we used the same OTT along the same camera characteristics used in Sec. 5.5. We fixed $\theta_2 = 0°$ and we varied distance from OTT to the camera from 30 cm to 200 cm in steps of 1 cm. For this experiment a total of 171 images were generated, some of the generated images are shown in Fig. 14.

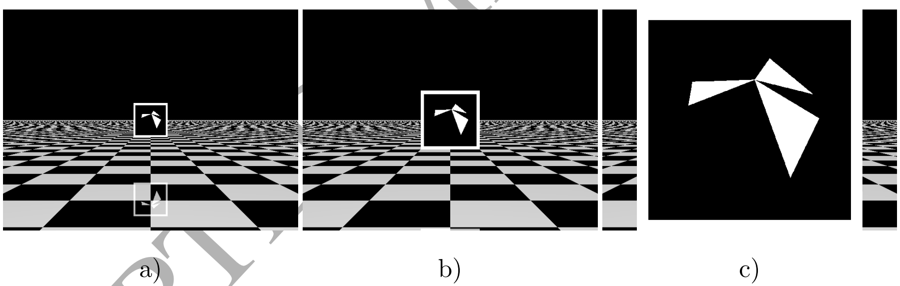


Figure 14: Instances of the ray-traced images for the distance experiment. a) Tag to 200 cm of distance. b) Tag to 115 cm of distance. c) Tag to 30 cm of distance.

From the 171 processed images 91 of them were successfully recovered. We detected the OTT in all 171 images but point set estimation was not correct in all cases thus ID was also affected. We observed that 100% of failures happened with $d \geq 91$ cm.

6. Discussion

We consider that the OTTs design described in Sec. 4 provides multiple advantages over other approaches. First, OTTs have an invariant ID codification and decodification mechanism. Second, all existing visual fiducial tags with squared border compute the pose from the four points of the border. Although four are the minimum of points to estimate the pose, OTTs can take advantage of their design to use also as matching features the vertices of the triangles inside to estimate the pose with more than four points. For this second point the ID computation algorithm can be modified to match features between the tag and the stored vertices position in the dictionary. This modification would increase the precision in the pose estimation and also would provide of more information like the reprojection error, i.e. the sum of error computed from the current vertices position and the computed with the homography.

From results in Sec. 5.1 we observe that OT maintains invariant to the transformations involved in the image generation process. Results show that OT is unaffected up to $\theta_2 = 80^\circ$ and $\theta > 65$ of tilt for C^7 and C^8 respectively. We consider this result as the justification for using OT as the base for the proposed tags.

From the complementary results in Sec. 5.2 we also observe that OT maintains unchanged to distance variations. Since point sets are given in a 256×256 grid, we can infer that OT will maintain unaffected as long distance does not reduce the point set resolution more than that size. The performance of this experiment, and also of the presented in Sec. 5.1, would enhance if the resolution of the camera is increased. In the experiment we used a 640×480 resolution for stressing the process but modern devices, like mobile phones, are equipped much higher camera resolutions (up to dozens of megapixels).

In our experiments we used point sets with 8 bits of precision. For point sets with 16 bits, theoretically a resolution of $2^{16} \times 2^{16}$ pixels would be required, although we have not verified that all 16 bits resolution is needed for C^9 and C^{10}.

The results of the experiment in Sec. 5.3 allow to analyze the effects of noise coming from several sources. Errors can come from many sources like from point set estimation, surface bending or even camera distortion. The experiment allows to generalize the analysis in such way that the effects of noise are observed in an analytic way. From Tab. 7 we observe that MP is not the same for all instances of a same group C^k, some C_l^k instances have greater MP than the minimal for the group. We see this as a positive aspect since those applications that require of few different OTTs can choose those point sets instances with bigger MP value. In addition, the results in Tab. 7 would linearly enhance with the resolution of the camera.

The results in Tab. 7 are the number of point sets instances with a determined MP value. These results depend on the instances of OT used. In this sense, we believe that by maximizing the MP of the point sets in C^k, we can obtain an optimized OT database with enhanced MP values and thus less sensible to noise effects.

With the proposed structure in Sec. 4 the maximum number of tags that can be generated is 3472, summing all the instances of $C^k, k \leq 8$. This number can be doubled if the background and data area tones are switched from black to white and vice versa in such way that a white data area defines 3472 valid tags and the black case the other 3472 as applied in the ARTags approach [13].

Observations in Secs. 5.4 and 5.5 show that OTTs work as expected. In 5.4 we probe the proposed tags invariance to rotations and perspective. We correctly recovered the ID from images with up to $\theta_2 = 81^\circ$ of rotation and perspective. In 5.5 we see that the ID was correctly recovered at distances less than 62 cm. As suggested in the previous paragraph, the maximum distance will of this experiment would increase as the camera resolution does.

The results of this study show that OT is a good mechanism for encoding ID in visual fiducials. Owing the OT properties we consider it as potentially useful to solve other problems like pattern recognition and feature matching. Specially for those cases when images are taken with different point of view.

Since OTT do not require any reference for estimating the orientation nor the size of the elements that conform the tag, other more aesthetic designs can be used for publicity purposes.

7. Conclusions and future work

This paper reports on the development of a visual fiducial tag based in OT. As main results we show that Order Type Tags:

- Are very robust to rotation, tilt and perspective transformations.
- They avoid the homography estimation that is a crucial step in the existing visual tags.
- Allow to generate up 3472 different tags but the number can be doubled with small considerations in the OTT structure.

As future work we aim to enhance the OTT point estimation phase to make it robust to occlusions and we also aim to develop an optimized OT database that maximizes MP for the results in Tab. 7.

[1] A. Kelly, Mobile Robotics: Mathematics, Models, and Methods, Cambridge University Press, 2013.

[2] I. Rabbi, S. Ullah, A survey on augmented reality challenges and tracking, Acta Graphica Journal Faculty of Graphic Arts University of Zagreb 24 (1-2) (2013) 29–46.

[3] R. S. Patkar, S. P. Singh, S. V. Birje, Marker based augmented reality using android os, International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) 3 (5) (2013) 64–69.

[4] R. Sun, Y. Sui, R. Li, F. Shao, The design of a new marker in augmented reality, in: Proc. Int. Conf. on Economics and Finance Research, 2011, pp. 129–132.

[5] N. Thiengtham, Y. Sriboonruang, Improve template matching method in mobile augmented reality for thai alphabet learning, International Journal of Smart Home 6 (3) (2012) 25–32.

[6] S. Siltanen, V. teknillinen tutkimuskeskus, Theory and Applications of Marker-based Augmented Reality, VTT science, 2012.

[7] B. Atcheson, F. Heide, W. Heidrich, Caltag: High precision fiducial markers for camera calibration., in: R. Koch, A. Kolb, C. Rezk-Salama (Eds.), VMV, Eurographics Association, 2010, pp. 41–48.

[8] A. Babinec, L. Jurišica, P. Hubinský, F. Duchoň, Visual localization of mobile robot using artificial markers, Procedia Engineering 96 (2014) 1 – 9. doi:10.1016/j.proeng.2014.12.091.

[9] R. Muñoz-Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, R. Medina-Carnicer, Mapping and localization from planar markers, Pattern Recognition 73 (2018) 158 – 171. doi:10.1016/j.patcog.2017.08.010.

[10] H. Kato, K. T. Tan, D. Chai, Barcodes for Mobile Devices, 1st Edition, Cambridge University Press, New York, NY, USA, 2010.

[11] P. D. Virulkar, A. N. Bhute, Comparative study: Location based mobile advertisement publishing system, in: Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on, 2015, pp. 570–574. doi:10.1109/ICCUBEA.2015.200.

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

[13] M. Fiala, Artag, a fiducial marker system using digital techniques, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 2, 2005, pp. 590–596 vol. 2. doi:10.1109/CVPR.2005.74.

[14] J. Heikkila, Geometric camera calibration using circular control points, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1066–1077. doi:10.1109/34.879788.

[15] M. Fiala, Designing highly reliable fiducial markers, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1317–1324. doi:10.1109/TPAMI.2009.146.

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

[17] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (6) (2014) 2280 – 2292. doi:10.1016/j.patcog.2014.01.005.

[18] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using mixed integer linear programming, Pattern Recognition 51 (2016) 481 – 491. doi:10.1016/j.patcog.2015.09.023.

[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, pp. 85–94. doi:10.1109/IWAR.1999.803809.

[20] Y. Li, Y.-T. Wang, Y. Liu, Fiducial marker based on projective invariant for augmented reality, Journal of Computer Science and Technology 22 (6) (2007) 890–897. doi:10.1007/s11390-007-9100-0.

[21] L. Teixeira, M. Loaiza, A. Raposo, M. Gattass, Augmented Reality Using Projective Invariant Patterns, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 520–529.

[22] F. Bergamasco, A. Albarelli, A. Torsello, Pi-tag: a fast image-space marker design based on projective invariants, Machine Vision and Applications 24 (6) (2013) 1295–1310. doi:10.1007/s00138-012-0469-6.

[23] A. Xu, G. Dudek, Fourier tag: A smoothly degradable fiducial marker system with configurable payload capacity, in: Proceedings of the 8th Canadian Conference on Computer and Robot Vision (CRV '11), St. John's, Newfoundland, Canada, 2011, pp. 40–47.

[24] J. Kim, P. Gurdjos, I. Kweon, Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration, IEEE Trans. PAMI 27 (4) (2005) 637–642.

[25] X. Yang, H. Qiao, Z.-Y. Liu, Point correspondence by a new third order graph matching algorithm, Pattern Recognition 65 (2017) 108 – 118. doi:10.1016/j.patcog.2016.12.006.

[26] J. Christmas, R. Everson, J. Bell, C. Winlove, Inexact bayesian point pattern matching for linear transformations, Pattern Recognition 47 (10) (2014) 3265 – 3275. doi:10.1016/j.patcog.2014.04.022.

[27] J. Tang, L. Shao, X. Zhen, Robust point pattern matching based on spectral context, Pattern Recognition 47 (3) (2014) 1469 – 1484, handwriting Recognition and other PR Applications. doi:10.1016/j.patcog.2013.09.017.

[28] S. Jung, S. Song, M. Chang, S. Park, Range image registration based on 2D synthetic images, Computer-Aided Design 94 (2018) 16 – 27. doi:10.1016/j.cad.2017.08.001.

[29] J. E. Goodman, R. Pollack, Multidimensional sorting, SIAM Journal on Computing 12 (3) (1983) 484–507.

[30] G. Aloupis, J. Iacono, S. Langerman, O. Özkan, S. Wuhrer, The complexity of order type isomorphism, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '14, SIAM, 2014, pp. 405–415.

[31] R. Johnsonbaugh, Discrete Mathematics, Pearson/Prentice Hall, 2009.

[32] O. Aichholzer, F. Aurenhammer, H. Krasser, Enumerating order types for small point sets with applications, Order 19 (3) 265–281, accessed: 2016-01-15. doi:10.1023/A:1021231927255.

[33] G. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, O'Reilly Media, 2008.

[34] A. C. Bovik, The Essential Guide to Image Processing, Academic Press, 2009.

[35] R. Hartley, A. Zisserman, Multiple view geometry, 1st Edition, Vol. 6, Cambridge university press, New York, NY, 2000.

[36] Persistence of vision raytracer., `http://www.povray.org/`, accessed: 2016-09-30.

Heriberto Cruz Hernández received the B.E. degree in Computational Systems Engineering from National Polytechnic Institute (IPN), Mexico, in 2010, and the M.Sc. degree in Computer Science from Center of Research and Advanced Studies (Cinvestav), Mexico, in 2012. He is now a Ph.D. student at Cinvestav, and his current research interest include Computer Vision and Evolutionary Computation.

Dr. Luis Gerardo de la Fraga received the B.S. degree in electrical engineering from Instituto Tecnológico de Veracruz, Veracruz, México in 1992; he received the M.Sc degree from the Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), Puebla, México, in 1994; and the Ph.D. degree from the Autonomous University of Madrid, Spain, in 1998. Since 2000 he is in the Computer Science Department at the Center of Research and Advanced Studies (Cinvestav), in Mexico City. He research areas include computer vision, optimization, and network security. He is very enthusiastic of open software and GNU/Linux systems.

Dr. de la Fraga has published more than 60 articles in journals and international conferences. He is member of ACM and IEEE societies since 2005.