

An Accurate and Robust Artificial Marker based on Cyclic Codes

Filippo Bergamasco, Andrea Albarelli, Luca Cosmo,
Emanuele Rodolà, *Member, IEEE*, and Andrea Torsello, *Member, IEEE*

Abstract—Artificial markers are successfully adopted to solve several vision tasks, ranging from tracking to calibration. While most designs share the same working principles, many specialized approaches exist to address specific application domains. Some are specially crafted to boost pose recovery accuracy. Others are made robust to occlusion or easy to detect with minimal computational resources. The sheer amount of approaches available in recent literature is indeed a statement to the fact that no silver bullet exists. Furthermore, this is also a hint to the level of scholarly interest that still characterizes this research topic. With this paper we try to add a novel option to the offer, by introducing a general purpose fiducial marker which exhibits many useful properties while being easy to implement and fast to detect. The key ideas underlying our approach are three. The first one is to exploit the projective invariance of conics to jointly find the marker and set a reading frame for it. Moreover, the tag identity is assessed by a redundant cyclic coded sequence implemented using the same circular features used for detection. Finally, the specific design and feature organization of the marker are well suited for several practical tasks, ranging from camera calibration to information payload delivery.

Index Terms—RUNE Tag, Fiducial Markers, Cyclic Codes, Camera Calibration, Pose Estimation

1 INTRODUCTION

AN artificial marker is a physical artifact consistent with a given model which can be used whenever a reliable pose estimation and identification is sought. This is the case, for instance, for many Computer Vision tasks ranging from pose recovery [1], [2], [3] to intrinsic [4] and extrinsic camera calibration [5]. Within these scenarios, the adoption of an artificial marker is often preferred over less invasive alternatives, such as features that can be naturally found within the scene. In fact, while more convenient from a practical point of view, natural features are not guaranteed to be abundant in every scene or to exhibit an adequate level of detection and recognition reliability. Furthermore, since an artificial marker can be used to satisfy different needs, it is valuable to be able to create application-specific designs. For these reasons, fiducial tags are not only a widely used tool in practice, but they are also a lively research topic. Since a marker is usually created to be easily detected by a pinhole-modeled camera, most approaches are designed to exploit the projective invariance of basic geometrical entities. Specifically, most markers that can be found in literature are based on projective-invariant features that are both simple and easy to detect, such as points, lines, planes and conics. While it is difficult to track back to the earliest marker designs, it is sensible to believe that circular dots were among the first shapes used. In fact, circles appear as ellipses under projective transformations and the associated conic is invariant with respect to intrinsic or extrinsic parameters of the camera. This allows a

fast and robust detection of the features directly on the image plane. Moreover, it is quite straightforward to find a proper homography which transforms back the found ellipses in circles, yielding an orthogonal view of the marker itself.

Such properties are exploited, for instance, in the earlier conception proposed by Gatrell [6], adopting a set of highly contrasted concentric circles which, after detection, are validated using the correspondences between the centroids of the ellipses found. In addition to the sheer feature localization, this design also allow to attach to each marker some additional information payload. This is obtained by alternating white and black circles according to some predefined pattern. This design has been slightly enhanced in [7] where the concentric circles are drawn using different colors and multiple scales, thus allowing to embed more information. Dedicated “data rings” are added to the fiducial design in [8] and [9]. A set of four circles located at the corner of a square is proposed in [10], where an additional pattern is placed between the four dots in order to distinguish between different targets. This ability to separate a set of different markers is crucial for complex scenes where more than a single fiducial is required to better handle occlusion or to track several objects at the same time. As an additional bonus, the availability of a coding scheme can be used to enable a validation step and to lower the number of false positives. For these reasons, a lot of effort has been dedicated to the design of effective coding schemas (see for instance [11], [12], [13]). A rather different but extensively used approach for marker recognition is to leverage on

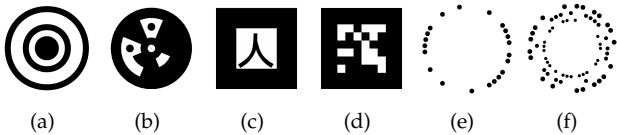


Fig. 1: Some examples of fiducial markers that differ both for the detection technique and for the pattern used for recognition. In the first two, detection happens by finding ellipses and the coding is respectively held by the color of the rings in Concentric Circles (a) and by the appearance of the sectors in Intersense (b). While ARToolkit (c) uses image correlation to differentiate markers, ARTag (d) relies on error-correcting binary codes. Finally, in (e) and (f) we show two instances of RUNE-43 and RUNE-129 respectively.

the geometrical properties of the cross ratio among detected feature points [14], [15], [16] or lines [17]. An interesting advantage of the cross ratio is that, being projective invariant, the recognition can be made without the need of any rectification of the image. Unfortunately, this comes at the price of a low overall number of distinctively recognizable patterns, and thus concurrently usable markers. In fact the cross ratio is a single scalar with a strongly non-uniform distribution [18] and this limits the number of well-spaced different values that can possibly be generated. Finally, also lines are a frequently used feature in the design of fiducial markers. Usually, they are exploited by detecting the border edges of a highly contrasted quadrilateral block. This is the case, for instance, for the ARToolkit [19] system which is often adopted as a reference baseline since it has a wide user base and its code is freely available in source form. Due to the easiness of detection and reasonable pose recovery accuracy that can be obtained with this kind of design [20], similar approaches are found in many recent proposal, such as ARTag [21] and ARToolkitPlus [22]. With these latter methods, the recognition technique of ARToolkit, which is based on image correlation between arbitrary images, is replaced by the reading of a binary coded pattern (see Fig. 1). The adoption of an error-correcting code makes both the marker detection and identification very robust, in fact we can deem these designs as the most successful from an applicative point of view.

With this paper we are introducing a novel fiducial marker that combines several strengths of different approaches, resulting in an all-rounder that can be directly applied in many scenarios. The key idea underlying our design is to entrust the robustness of the detection process to a well-grounded and occlusion-resilient cyclic code rather than to the geometrical features themselves. While this is not the first attempt in this direction [23], our proposal introduces some notable features not available with previous designs. For starters, the marker we are introducing is arranged to facilitate its localization thanks to a reading frame that can be fully constrained using only

two marker dots. Furthermore, our design is flexible enough to allow to use different amounts of dots, granting a higher robustness or a greater working distance, depending on the specific scenario. Finally, the large number of dots provided by our marker, beside boosting the pose estimation accuracy, can be exploited to enable applications that are usually considered beyond the domain of fiducial markers, such as camera calibration.

The remaining of the paper is organized as follows. In section 2 we describe the general design and we introduce two localization methods, to be used respectively with calibrated and uncalibrated cameras. Afterwards, we introduce the adopted coding strategy and we suggest a technique to perform instant decoding, including proper recovery from errors due to occlusion and misdetection of the marker dots. In section 3 we test the accuracy achieved when dealing with different real-world problems and we compare the obtained performance with some widely used fiducial markers.

2 RINGS OF UNCONNECTED ELLIPSES

We design our tags as a set of circular high-contrast features (*dots*) spatially arranged into concentric *layers* (See Fig. 2). The tag internal area, delimited by the outer layer, is divided into several evenly distributed circular *sectors*. Each layer and sector intersection defines a *slot* that may or may not contain a dot.

In a tag composed by m layers and n sectors, we can encode a sequence of n symbols taken from an alphabet of 2^m elements. Each element of the alphabet is simply defined as the binary number encoded by the presence or absence of a dot. For example, if the 14th sector of a 3-layer tag contains a dot in the first and the last layer, we encode the 14th symbol with the number $5_{10} = 10_{12}$. In this paper we propose two instances of such design, namely *RUNE-43* and *RUNE-129*. The first is composed by a single layer divided into 43 sectors. Since the alphabet contains

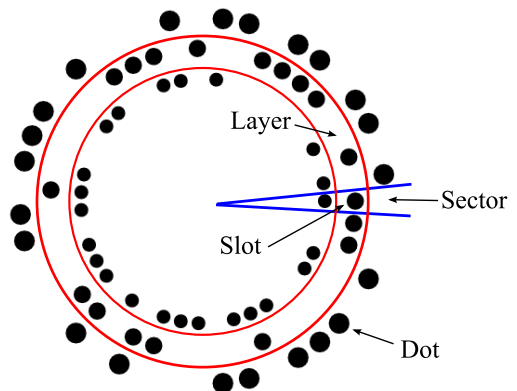


Fig. 2: Our proposed design divided into its functional parts. An instance of a 3-layers RUNE-129 is displayed.

only 2 elements (1 bit given by the presence or absence of a dot), each *RUNE-43* encodes a sequence of 43 binary symbols. Conversely, the latter is composed by 3 layers divided into 43 sectors. 3 slots for each sector allow to encode a sequence of 43 symbols from an alphabet of $2^3 = 8$ elements. Not surprisingly, not all the possible codes can be used as valid codes for the tag. For instance, the tag composed by only empty slots does not make any sense. Therefore, we require the coding strategy to respect some properties to uniquely identify each dot regardless the projective transformation involved. We discuss this topic in detail in section 2.3.

Finally, we set the dot radius equals to κ -times to the radius of the layer at which the dot is placed. We can take advantage of this property to dramatically speed up the detection as explained in section 2.1.

2.1 Candidate selection with a calibrated camera

One of the core features of every fiducial marker system is its ease of detection. Even if one of our principles is to promote the accuracy over the speed, we still need to setup an efficient way to identify each circular feature among the tags. Given an image, we start by extracting a set of candidate dots. To do this, we use a combination of image thresholding, contour extraction and ellipse fitting provided by the OpenCV library [24]. Additionally, a subsequent naive filtering step based on dot eccentricity and area keeps only whose features respect a reasonable prior. Finally, each extracted ellipse can be further refined by using common sub-pixel fitting techniques such the one proposed in [25]. We give no additional details on the specific procedure we follow since is not important for all the subsequent operations. Any suitable technique to extract a set of circular dots from a generic scene would be fine.

At this point, we need a method to cluster all the candidate dots into different possible tags and discard all the erroneous ones that were originated by noise or clutter in the scene. Since we know that the dots are arranged in circular rings, we expect that dots belonging to the same layer would appear disposed around an elliptical shape once observed through a central projection. Therefore, dots in the same layer can be identified by fitting an ellipse through their 2D image coordinates and verifying the distance assuming this model.

Total ellipses	10	50	100	500
Exhaustive search	252	2118760	75287520	$> 10^{10}$
Proposed method	45	1225	4950	124750

TABLE 1: Number of RANSAC tests required for an exhaustive search of all ellipses that can be fitted given each subset of 5 dot candidates found in an image.

A common approach would consist in the use of a RANSAC scheme that uses a set of 5 candidate dots to estimate the model (i.e. the ellipse) against which quantify the consensus of all the others. Unfortunately, since 5 points are needed to characterize an ellipse into the image plane, the use of RANSAC in a scenario dominated by false positives (even without clutter we expect the majority of dots to belong to different tag or even layer) would quickly lead to an intractable problem. In Table 1 we give a quick comparison between the number of model tests required by our method against an exhaustive search of all possible 5-dots subsets. A possible alternative could be the use of a specialized Hough Transform [26], but also this solution would not be effective since hindered by the relative low number of samples and the high dimensionality of the parameter space.

What makes possible the detection of our tags in reasonable time is the observation that there exists a relation between the shape of a dot and the shape of the ring in which is contained. Specifically, they both appear as two ellipses (since they originate from a projection of two circles) and the parameters of both curves depend on the relative angle between the camera and the plane in which they lie. Even if from a single conic is not possible to recover the full camera pose, there is still enough information to recover (up to a finite set of different choices) a rotation that transform that conic into a circle. This, combined with a known relation between the relative size of the dots and the rings, can give clues of the geometry of a layer and so ease the clustering process.

In this section, we give a detailed description on how the recovering of such rotation is done assuming a known camera matrix. In many situations, the requirement of a calibrated camera is not particularly limiting. For instance, if our tags would be used as a coarse registration method for a structured-light scanner solution (we give examples of this in section 3), the camera would certainly be calibrated as implied by the reconstruction process. However, for the high accuracy exhibited in points localization, it would be interesting to use our tags as a calibration target instead of a classical chessboard. To deal with this situations, we propose a way to still use our tags in an uncalibrated scenario in section 2.2.

Given the set of initial dot candidates, we start by recovering the parameters describing their elliptical shape. Specifically, we translate the image reference frame so that the principal point coincides with the origin, and parametrize each conic as the locus of point such that:

$$\vec{x}^T \mathbf{Q} \vec{x} = \begin{pmatrix} u & v & 1 \end{pmatrix} \begin{pmatrix} A & B & -\frac{D}{f} \\ B & C & -\frac{E}{f} \\ -\frac{D}{f} & -\frac{E}{f} & -\frac{F}{f^2} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \quad (1)$$

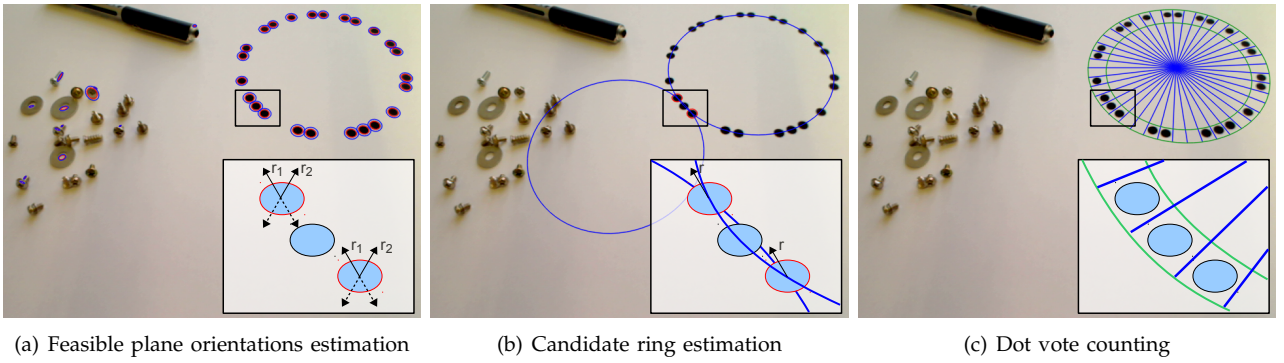


Fig. 3: Steps of the ring detection: in (a) the feasible view directions are evaluated for each ellipse (with complexity $O(n)$), in (b) for each compatible pair of ellipses the feasible rings are estimated (with complexity $O(n^2)$), in (c) the dot votes are counted, the code is recovered and the best candidate ring is accepted (figure best viewed in color).

Where f is the camera focal length and u, v are pixel coordinates.

We follow [27] to estimate a rotation around the optical center that transforms the ellipse described by \mathbf{Q} into a circle. Specifically we decompose \mathbf{Q} via SVD

$$\mathbf{Q} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \quad \text{with } \mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$$

and compute the required rotation as:

$$\mathbf{R}_{\mathbf{Q}} = \mathbf{V} \begin{pmatrix} g \cos \alpha & s_1 g \sin \alpha & s_2 h \\ \sin \alpha & -s_1 \cos \alpha & 0 \\ s_1 s_2 h \cos \alpha & s_2 h \sin \alpha & -s_1 g \end{pmatrix} \quad (2)$$

$$g = \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}}, \quad h = \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}}$$

where s_1 and s_2 are two free signs, leaving 4 possible matrices and α is any arbitrary rotation around the normal of the plane which remains constrained while observing just a single ellipse. At this point, if we fix $\alpha = 0$, each detected ellipse \mathbf{Q} may spawn four different rotation matrices $\mathbf{R}_{\mathbf{Q}}^i, i = 1 \dots 4$ that transforms the conic into a circle.

Since two of this four candidates imply a camera observing the back-side of the marker, we can safely discard all the $\mathbf{R}_{\mathbf{Q}}^i$ for which the plane normal $N_{\mathbf{Q}}^i = \mathbf{R}_{\mathbf{Q}}^i \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T$ is facing away from the camera (i.e. the last component is positive).

At this point, we search for whole markers by simultaneously observing the rotation matrices of a couple of ellipses. Specifically, for each pair \mathbf{Q}_k and \mathbf{Q}_w , we produce the set of the four possible rotation pairs $\mathfrak{R} = \{(\mathbf{R}_{\mathbf{Q}_k}^i, \mathbf{R}_{\mathbf{Q}_w}^j); i, j = 1 \dots 2\}$. From this set, we remove the pairs for which the inner product of the relative plane normals is below a fixed threshold and average the remaining rotation pairs by means of quaternion mean. Finally, we keep the best rotation average by choosing the one that minimize the difference between the radii of \mathbf{Q}_k and \mathbf{Q}_w after being transformed by such rotation. The rationale is to avoid to choose ellipses with discordant orientations (as the

marker is planar) and to use a compatibility score that takes advantage of the fact that ellipses on the same ring should be exactly the same size on the rectified plane.

Whenever a pair of dots \mathbf{Q}_k and \mathbf{Q}_w generate a good average rotation $\mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}$, two hypothesis on the ring geometry can be made (Fig. 3.b). Indeed, we expect the ring shape being such that the following two properties holds. First, it should pass through the centers of \mathbf{Q}_k and \mathbf{Q}_w . Second, the ratio between the ring radius and the radii of \mathbf{Q}_k and \mathbf{Q}_w , after being transformed through $\mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}$, should be exactly κ . Operatively, we first fit the two circles $\mathbf{C}_1, \mathbf{C}_2$ passing through the centers of $\mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}^T \mathbf{Q}_w \mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}$ and $\mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}^T \mathbf{Q}_k \mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}$ and having radius $\kappa \hat{r}$ where \hat{r} is the average radius of the two transformed dots. Then, we transform \mathbf{C}_1 and \mathbf{C}_2 back through the inverse of $\mathbf{R}_{(\mathbf{Q}_k, \mathbf{Q}_w)}$.

As soon as candidate rings are extracted, a circular grid made by sector and layers can be generated directly on the image (Fig. 3.c). Of course, if the tag is composed by more than one layer, we need to generate additional rings both inward and outward. Then, for each slot the presence or absence of a dot can be observed to produce a binary sequence that will be analyzed in the decoding step to identify or discard the candidate marker.

To summarize, the detection step goal is to identify possible markers candidates by searching groups of dots belonging to the same ring, expecting them arranged in an elliptical shape. To do so, we avoid the direct estimation of ellipses in the image since it would require an unfeasible effort. Diversely, we take advantage of the geometrical properties of the dots and the known ratio κ to obtain two possible ring candidate for each pair of ellipses. As result, only $O(n^2)$ operations are required.

2.2 Dealing with the uncalibrated case

The approach described so far assumed a calibrated camera setup. Indeed, all the rotation matrices $\mathbf{R}_{\mathbf{Q}}$

were designed to transform conics lying on the normalized image plane (hence requiring the focal length) around the camera optical center. It has to be noted, however, that camera intrinsics are not an implied requirement of the tag itself but just a powerful tool to dramatically speed-up the detection. As a consequence, we would be satisfied to just guess a raw estimation of the focal length and principal point good enough to still produce fair rotation matrices and sustain the detection procedure.

We decided to use the image center as our guess of the principal point. Even if it appears a bold assumption, we observed that this holds for most cameras. Diversely, the focal length is difficult to guess as it depends on the lens mounted. However, also in this case we can take advantage on the geometric properties involved when observing a set of coplanar circles.

In Section 2.1 we discussed how two feasible plane normals can be estimated from each conic. It's crucial to observe that, if we apply the same projective transformation to two circles lying on the same plane, only one plane normal estimated from the first circle will be parallel to a normal extracted from a second, whereas the other two will diverge [28]. Furthermore, this property holds only for the correct focal length and principal point and can be naturally expanded to multiple coplanar conics.

To better explain this behaviour, we extracted all orientations from a set of 3 coplanar circles assuming to know the principal point and varying the focal length. In fig. 4 (Left) we plotted the values of such orientations in spherical coordinates spanning positive values of f from almost zero to 5 times the known correct value. For the right plot we did the same procedure but with negative values. In general, each ellipse produces two different traces in (ϕ, θ) -plane as a function of the focal length. Since all correct orientations have to be parallel to each other when the correct focal length is used, traces that are relative to the correct orientation will converge to a same

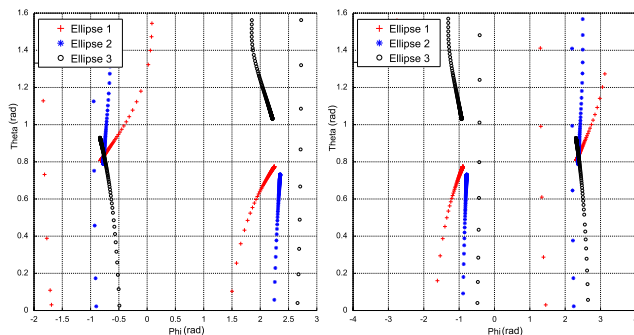


Fig. 4: Estimated normals orientation in spherical coordinates of three coplanar ellipses spanning positive (Left) and negative (Right) focal length values. Note how one of the two possible orientations converge to a common direction while the other does the opposite.

point as f get closer to the expected value. On the other hand, all other traces will follow different paths and will diverge to different directions. It's clear from the plot that for positive values of the focal length there is only one intersection point (in this example $\phi \simeq -0.76, \theta \simeq 0.83$). Also, since the other possible intersection only happens when f becomes negative, the wrong orientation will never be present in the set of feasible solutions.

This means that we can both estimate the correct focal length and extract sets of coplanar circles by solving a clustering problem among all the generated plane normals. However, there is no simple closed form solution to reverse the process and obtain the best possible focal length that would have produced a given normal. Therefore, we restrict our estimation to a discrete set of n_f possible focal length values $f_i, i = 1 \dots n_f$ equally spaced inside the range $f_{\min} \dots f_{\max}$. At this point, for each dot \mathbf{Q} detected in a scene and for each f_i , exactly two feasible plane normals $N_{\mathbf{Q}_{f_i}}^1, N_{\mathbf{Q}_{f_i}}^2$ can be computed as described in section 2.1. All such normals will exhibit two degrees of freedom and hence can be easily parametrized in spherical coordinates with azimuth ϕ and elevation θ as vectors in \mathbb{R}^2 . Then, all these vectors are collected in a 2D accumulator whose bins are equally divided into equal angular ranges.

Once the accumulator is completely filled with values extracted from all the dots, local maxima with respect of the cardinality of the bins will represent clusters of normals oriented almost in the same direction. Finally, once a maxima is selected, we take the median focal length of all the candidates contained in a bin as our sought focal length estimate. Moreover, the candidates contained in a winning bin are all coplanar and thus the dots search phase can be restricted on such set.

An example of the proposed focal length estimation method is given in Fig. 5. We started by rendering a synthetic image of a RUNE-149 tag trough a virtual camera of known focal length $f_{vc} = 1000 px$ and with principal point being exactly the center of the image (First row of Fig. 5). In the middle of the figure, we plotted the accumulator values projected on the front and back side of a hemisphere. As expected, a clear accumulation of votes can be observed in the bin containing the combination of ϕ, θ corresponding to the normal of the plane on which the tag lie. On the right, we plotted the distribution of the focal length candidates of the winning bin, highlighting a clear maximum around the correct value of f_{vc} . Conversely, we repeated the same test with two tags on the same image lying into two different planes (Second row of Fig. 5). This time, the accumulator shows two clear maxima corresponding to the plane normals of the two planes. Again, on the right side of the figure we plotted the distribution of the focal length candidates for the two winning bins. Two important observations

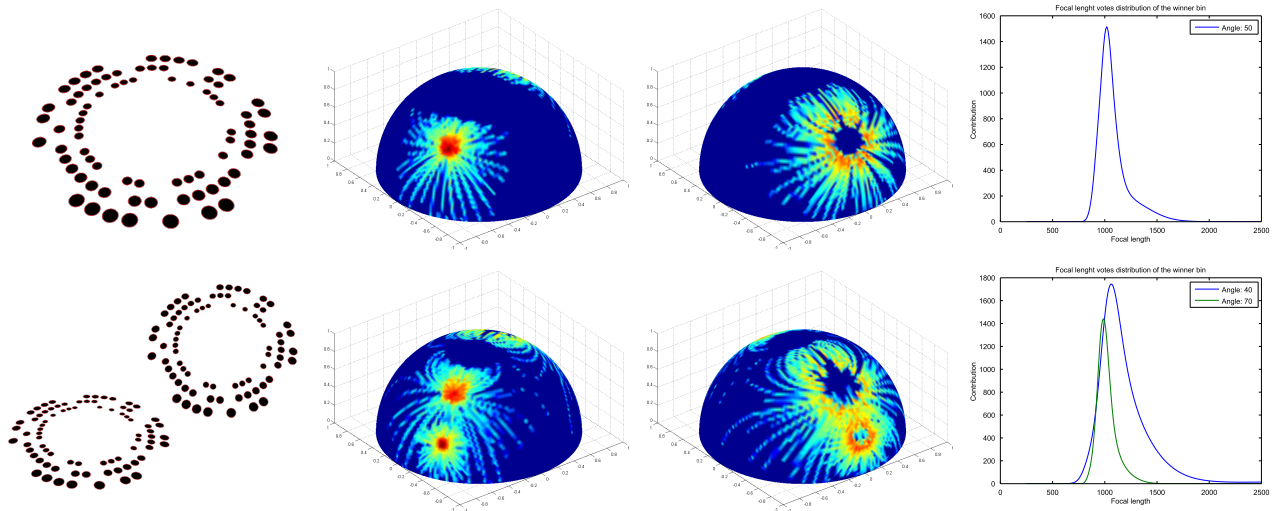


Fig. 5: A synthetic representation of the marker dots normal voting scheme used to guess an initial value of the camera focal length. Left: RUNE-129 markers rendered by a virtual camera with known focal length and principal point. Center: the normal accumulator visualized on the unitary sphere. Right: Focal length distribution of the bins. See the text for a complete discussion on the voting procedure.

can be made. First, both the two distributions show two clear maxima around f_{ver} demonstrating that a focal length guess is the same regardless of the tag orientation. Second, the more a tag is angled the more the guess is near the expected value. This can be explained by noting that a tag perfectly parallel to the imaging plane has all the dots exactly appearing as circles and so no focal length can be recovered. Therefore, the correct focal length is better constrained when the eccentricity of the dots is low. In fact, from the accumulator can be noted that the maximum corresponding to the angled tag is far more sharp than the other.

Even if the focal length guess is somehow biased by the angle of the observed tag, we feel that this won't be a show-stopper as we can still obtain a focal length guess good enough to let the detection procedure work properly. To convince the reader furthermore, we recall that the focal length is used to obtain a good rotation matrix to transform all the dots into circles. The more the angle is low, the more the focal length become irrelevant to recover that rotation. In the extreme case, to detect a perfectly parallel tag the focal length is not necessary at all since all the dots (and so the whole tag) already appear as circles.

To conclude, in the uncalibrated case we require an initial camera intrinsic parameters guessing step able to produce values good enough to perform a subsequent tag detection. To do so, we guess the principal point as the image center and the focal length with a voting procedure among a discretized set of plausible focal length values.

2.3 Marker Recognition and Coding Strategies

Once a candidate marker has been detected, dots distribution among the slots produces a sequence of

symbols that can be subsequently used to identify each tag. However, two coupled problems raise. First, we don't have a starting position of the symbols sequence since the detection step can only identify each candidate up to a rotation around the normal of the plane¹. Consequently, any cyclic shift of the sequence is equally possible and must be recovered. Second, some dots may be missing or assigned to wrong slots thus requiring the identification procedure being somehow robust to this situations.

We decided to cast the problem into the solid mathematical framework of coding theory. Specifically, dot patterns of the tags corresponds to codes generated with specific properties and error-correcting capabilities. In section 2.3.1 we briefly discuss the mathematical theory involved in the generation of the codes while in section 2.3.3 we give a closed form solution to decode each code block in case of erasures and errors. We refer the reader to [29] for a in-depth investigation of the field.

2.3.1 Code generation

We start by defining a *block code* of length n over a set of symbols S as the set $C \subset S^n$. The elements of C are called *codewords*.

Let $q = p^k \in \mathbb{N}$ be a power of a prime number p and an integer $k > 1$. We denote with \mathbb{F}_q the finite field with q elements. A *linear code* C is a k -dimensional vector sub-space of $(\mathbb{F}_q)^n$ where the symbols are taken over the field \mathbb{F}_q . A linear code is called *cyclic* if any cyclic shift of a codeword is still a codeword, i.e.

$$(c_0, \dots, c_{n-1}) \in C \Rightarrow (c_{n-1}, c_0, \dots, c_{n-2}) \in C$$

1. Note that, conversely, the verse of the sequence is induced by the counter-clockwise ordering of the sectors that is preserved since we always observe the frontal face of the marker plane.

If we consider the field $\mathbb{F}_q[x]/(x^n - 1)$ obtained by the polynomial ring $\mathbb{F}_q[x]$ modulo division by $x^n - 1$, there exists a bijection to the vectors in $(\mathbb{F}_q)^n$:

$$(v_0, \dots, v_{n-1}) \Leftrightarrow v_0 + v_1x + \dots + v_{n-1}x^{n-1}$$

Furthermore, C is a cyclic code if and only if C is an ideal of the quotient group of $\mathbb{F}_q[x]/(x^n - 1)$. This means that all cyclic codes in polynomial form are multiples of a monic *generator polynomial* $g(x)$ of degree $m < n$ which divides $x^n - 1$ in $\mathbb{F}_q[x]$. Since multiplying a polynomial form of a code by x modulo $x^n - 1$ corresponds to a cyclic shift

$$x(v_0 + v_1x + \dots + v_{n-1}x^{n-1}) \bmod (x^n - 1) = v_{n-1} + v_0x + \dots + v_{n-2}x^{n-2}$$

all codewords can be obtained by mapping any polynomial $p(x) \in \mathbb{F}_q[x]$ of degree almost $n - m - 1$ into $p(x)g(x) \bmod (x^n - 1)$.

Since all the cyclic shift are codes, we can group the codewords into *cyclic equivalence classes* such that two codewords are in the same class if and only if one can be obtained as a cyclic shift of the other. Since the number of elements in a cyclic equivalence class divides n , by choosing an n prime we only have classes either composed by a single element (constant codewords with n repetitions of the same symbol) or where all codewords are distinct. The first can be easily eliminated since it involves in at most q codewords.

In our marker setting, the identity of the marker is encoded by the cyclic equivalence class while the actual alignment of the circles (i.e. its rotation around the plane normal) can be obtained from the detected element within the class. Using coding theory enables us to balance the trade-off between the number of errors that can be handled with respect to the number of possible valid tags (i.e. the number of equivalence classes) granted. To our knowledge, is the first fiducial marker system that provides such feature at a geometrical level, modifying its shape to accommodate different requirements.

The *Hamming distance* $dH : S \times S \rightarrow \mathbb{N}$ is the number of symbols that differ between two codewords. Similarly, the Hamming distance of a code C is the minimum distance between all the codewords: $dH(C) = \min_{u,v \in C} dH(u, v)$. The Hamming distance plays a crucial role on the number of errors that can be detected and corrected. Indeed, a code with a Hamming distance d can detect $d - 1$ errors and correct $\lfloor (d-1)/2 \rfloor$ erasures. When we consider a linear code of length n and dimension k , the singleton bound $d \leq n - k - 1$ holds. Thus, with a fixed code length n the error correcting codes capabilities are traded with a smallest number of available codewords. In our setting we restrict our analysis to the correction of

random errors or erasures but the same mathematical framework can be used to improve the detection resilience while correcting burst errors (i.e. errors that are spatially coherent, like we have in case of occlusions).

For the proposed RUNE-Tags, we experiment on two specific codes instances. In the first one (RUNE-43) we encode the single-layer circular pattern as a vector in $(\mathbb{Z}_2)^{43}$, where \mathbb{Z}_2 is the remainder class modulo 2. The number 43 for the radial elements was chosen because it is a prime that leads to radial sectors of a reasonable size (slightly less than 10°), but any cyclic code of prime length would work. The polynomial $x^{43} - 1$ factors into 4 prime polynomial in \mathbb{Z}_2 , namely $x - 1$ and three polynomials of degree 14. By excluding $x - 1$ which generates only constant codes, and one of the prime polynomial of degree 14, we obtain a generator polynomial resulting in a cyclic code of dimension 15, with 762 different markers (equivalence classes) and a minimum Hamming distance of 13, allowing us to correct up to 6 errors. In particular, we used the polynomial (3) where the terms in brackets are two of the three degree 14 prime polynomials dividing $x^{43} - 1$.

$$g(x) = (1 + x^2 + x^4 + x^7 + x^{10} + x^{12} + x^{14})(1 + x + x^3 + x^7 + x^{11} + x^{13} + x^{14}) \quad (3)$$

In the second (RUNE-129) we have 8 different patterns (since it is a 3-layer tag) in a sequence of 43 sectors. We hold out the pattern with no dots to detect erasures due to occlusions and we encode the remaining 7 patterns as vectors in \mathbb{Z}_7 . For the whole target, the code is represented as vectors in $(\mathbb{Z}_7)^{43}$ using the generator polynomial (4).

$$g(x) = (1 + 4x + x^2 + 6x^3 + x^4 + 4x^5 + x^6)(1 + 4x^2 + 6x^3 + 4x^4 + x^6)(1 + x + 3x^2 + 5x^3 + 3x^4 + x^5 + x^6)(1 + 5x + 5x^2 + 5x^4 + 5x^5 + x^6)(1 + 6x + 2x^3 + 6x^5 + x^6)(1 + 6x + 4x^2 + 3x^3 + 4x^4 + 6x^5 + x^6) \quad (4)$$

Again, this polynomial is produced excluding one prime factor of $x^{43} - 1$. In particular, in \mathbb{Z}_7 , $x^{43} - 1$ factors into the usual $x - 1$ and 7 prime polynomials of degree 6. By excluding $x - 1$ and one of the degree 6 factors we obtain a cyclic code of dimension 7, giving 19152 different markers with a minimum Hamming distance of 30, and allowing us to correct up to 14 errors, or 29 erasure, or any combination of t errors and e erasures such that $2t + e \leq 29$. Unlike the case for (RUNE-43) where any choice of the prime factor to exclude leads to equivalent codes, here the choice of the prime factor to exclude was dictated by the need to have a generalized BCH code for fast decoding, as it will be explained in the next Section. Of the 6 prime polynomials, only 2 produced a BCH code correcting 14 errors, while the others had smaller correction capabilities through BCH decoding.

Once the locations of the errors are computed finding the roots of $\Lambda(x)$ among the powers of α , the error values are computed as follows:

$$e_j = -\frac{\alpha^{-(c-1)j}\Omega(\alpha^{-j})}{\Lambda'(\alpha^{-j})}, \quad (16)$$

where $\Lambda'(x) = \sum_{i=1}^{2t-1} i\lambda_i x^{i-1}$ is the formal derivative of $\Lambda(x)$.

In the case of the presence of e erasures, a BCH code can correct up to $t' = t - \lceil e/2 \rceil$ errors. First we set the erased digits to 0, and then we consider the *error/erasure locator polynomial* $\Gamma(x) = \Lambda(x)E(x)$ where $E(x)$ is the *erasure locator polynomial*

$$E(x) = \prod_{j=1}^e (1 - \alpha^{i_j} x). \quad (17)$$

In this context i_1, \dots, i_e are the indices of the e erasures.

With the error/erasure locator polynomial at hand, we re-write (14) as

$$\begin{aligned} \Omega(x) &= S(x)\Gamma(x) \pmod{x^{2t}} \\ &= S(x)E(x)\Lambda(x) \pmod{x^{2t}}, \end{aligned} \quad (18)$$

with $S(x)$ and $E(x)$ known, and $\Omega(x)$ and $\Gamma(x)$ of degree less than or equal to $t' + e$. As before, we use the Euclidean algorithm to compute $\Omega(x)$ and $\Lambda(x)$. Then, the roots of $\Gamma(x)$ will give us the locations of the errors, and their values are computed as follows:

$$e_j = -\frac{\alpha^{-(c-1)j}\Omega(\alpha^{-j})}{\Gamma'(\alpha^{-j})}. \quad (19)$$

It is easy to show that (RUNE-129) is a generalized BCH code correcting 14 errors, since $\alpha = x^5 + 4x^4 + 5x^2 + 6x$ defined over $\mathbb{Z}_7/(x^6 + 6x^5 + 2x^3 + 6x + 1) \simeq \mathbb{F}_{7^6}$ is a primitive 43-th root of unity, and the 28 consecutive powers $\alpha^8 \dots \alpha^{35}$ are all roots of (4). This means that through Forney's algorithm we can decode t errors and e erasures up to $2t + e \leq 28$, thus quite close to the code's limit.

In the case of (RUNE-43), the code cannot be reduced to a generalized BCH, however, $\alpha = x^7 + x^5 + x^4 + x^2 + x + 1$ defined over $\mathbb{Z}_2/(x^{14} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^4 + x^3 + 1) \simeq \mathbb{F}_{2^{14}}$ is a primitive 43-th root of unity, and has 3 ranges of 6 consecutive powers that are roots of (3), namely $\alpha^1 \dots \alpha^6$, $\alpha^{19} \dots \alpha^{24}$, and $\alpha^{37} \dots \alpha^{42}$. Limiting the correction capabilities to 5 errors, we have 8 Newton equations using only known syndromes. Consequently, we can solve the system of linear equations in $\mathbb{F}_{2^{14}}$ to find the error locator polynomial and, thus, the error locations. Since the code is binary, the error values are all 1 and we only need to flip the codes at the error locations. Note that we can detect whether the sequence has more than 5 errors by the fact that the locator polynomial has roots that are neither 0 nor powers of α , leaving us with a margin around the decision boundary of the codewords where we detect but cannot correct the

error. This is not a major problem in our application where it is arguably safer to ignore a tag with enough errors to be adjacent to the decision boundary.

2.3.4 Code Alignment

The alignment of the code is performed through an integer Fourier Transform. Let k be an integer such that $r = kn + 1$ is prime, and let α be a primitive element of the multiplicative group \mathbb{Z}_r^\times , i.e., $\alpha \neq 0$ and $\alpha^i \neq 1$ for all $i < r - 1$. Under these assumptions α^k has a prime period n . Assume also that for the given k , α combination $\alpha^k > q$. In our cases with $n = 43$ we have $k = 4$, $r = 4 \cdot 43 + 1 = 173$ prime, and 2 a primitive element of \mathbb{Z}_{173} . Further $2^4 = 16 > 7 > 2$.

Given a codeword sequence c_0, \dots, c_{n-1} of integers between 0 and $q - 1$, we define the Fourier transform in \mathbb{Z}_r of this sequence as

$$C_i = \sum_{j=0}^{n-1} \alpha^{kij} c_j \pmod{r}. \quad (20)$$

We define the phase of a Fourier coefficient C_i as $\phi_i = \log_\alpha(C_i)$, i.e., the unique integer $0 \leq \phi_i \leq r - 2$ such that $\alpha^{\phi_i} = C_i \pmod{r}$. With this definition we compute the shift of the code as $l = \lfloor \phi_1/k \rfloor$ and the Fourier Transform of the aligned code as

$$\hat{C}_i = \alpha^{-kli} C_i \pmod{r}. \quad (21)$$

Note that for constant codewords, $C_1 = \dots = C_{n-1} = 0$ so the shift is not well defined, while in all other cases we have $C_1 \neq 0$. Also, under this rotation $0 \leq \log_\alpha(\hat{C}_1) < k$, thus we are minimizing the phase of \hat{C}_1 . Once the aligned Fourier Transform is at hand, we can compute the aligned codeword sequence $\hat{c}_0, \dots, \hat{c}_{n-1}$ using the inverse Fourier Transform in \mathbb{Z}_r :

$$\hat{c}_i = n^{-1} \sum_{j=0}^{n-1} \alpha^{-kij} \hat{C}_j \pmod{r}. \quad (22)$$

3 EXPERIMENTAL VALIDATION

We tested our proposed fiducial markers in many different ways. To start, in section 3.1 and 3.4 we assessed the pose estimation accuracy compared to the ARToolkit [19] and ARToolkitPlus [22] which are deemed as a de-facto standard markers for augmented reality applications. Such tests are performed synthetically by rendering different frames varying an additive Gaussian noise, blur, illumination gradient and random occlusions. In all the tests, we simulated a 1024x768 pixel camera with a single tag placed at the image center spanning an area of 400x400 pixels.

Furthermore, driven by the good localization accuracy and occlusion resilience of the composing circular features, we tested RUNE-Tags as targets for camera calibration and object measurement. In section 3.2 we simulated a mono camera calibration scenario while in 3.3 we compared the camera pose estimation for both the mono and the stereo case. Also, we assessed the repeatability achievable while estimating the distance between two joint tags moving in a scene.

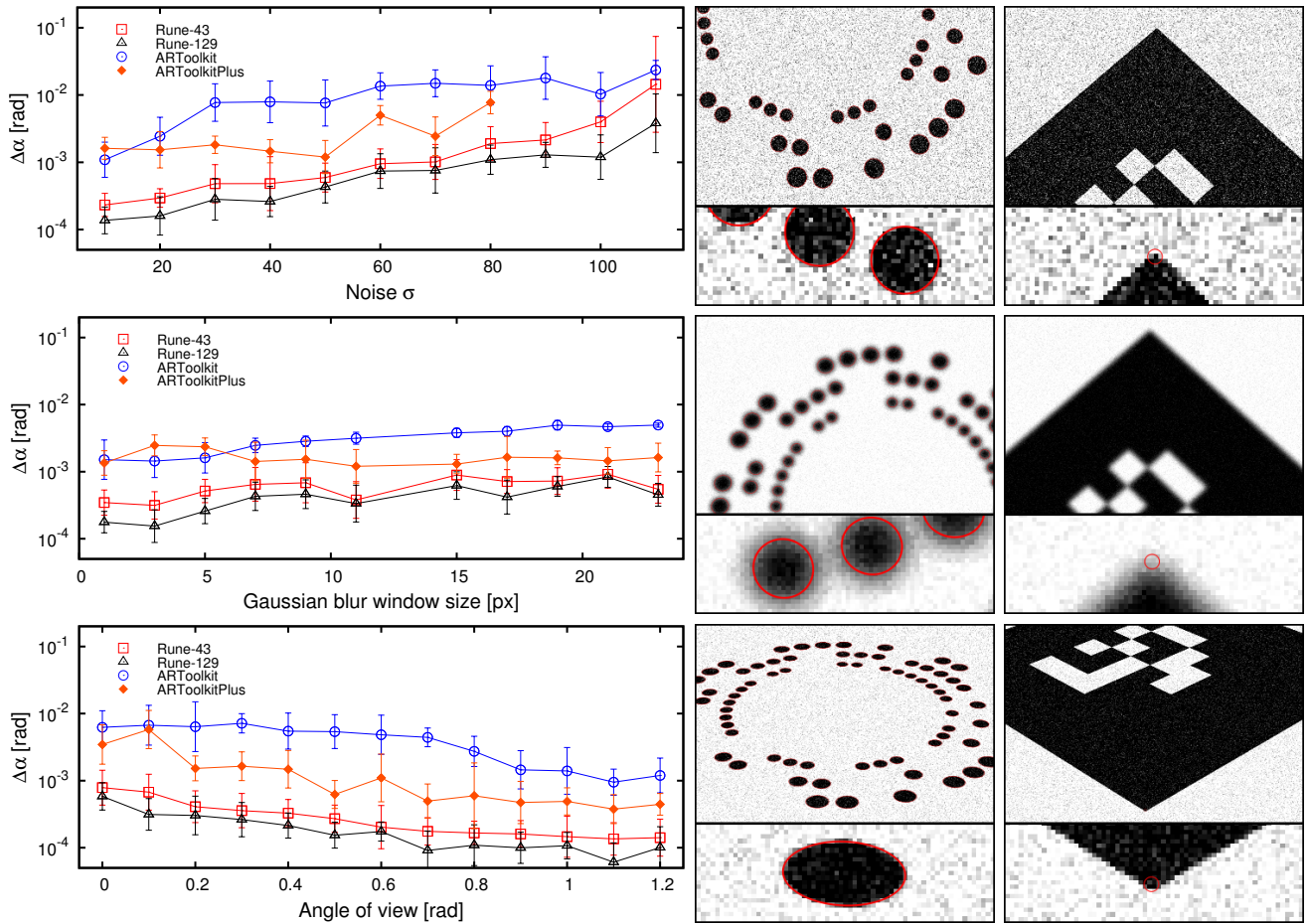


Fig. 6: Evaluation of the accuracy in the camera pose estimation with respect to different scene conditions. Examples of the detected features are shown for RUNE-129 (first image column) and ARTToolkitPlus (second image column).

Finally, in addition to the evaluation with synthetic images, in section 3.6 we performed some qualitative tests on real videos.

3.1 Accuracy and Baseline Comparisons

In Fig. 6 the accuracy of our markers with calibrated cameras is evaluated in terms of the relative pose angle $\Delta\alpha$ (the arc-cosine of the dot product between the last columns of the recovered rotation matrix and the ground truth). In the first test, an additive Gaussian noise was added to images with an average view angle of 0.3 radians wrt. marker plane normal and no artificial blur added. The performance of all methods get worse with increasing levels of noise and ARTToolkitPlus, while in general more accurate than ARTToolkit, breaks when dealing with a noise with a std. dev. greater than 80 (pixel intensities goes from 0 to 255). Both RUNE-43 and RUNE-129 always recover a more faithful pose. We think that this is mainly due to the larger number of correspondences used to solve the PnP problem. In fact, we can observe that in all the experiments RUNE-129 performs consistently better than RUNE-43.

Unlike additive noise, Gaussian blur seems to have a more limited effect on all the techniques. This is

mainly related to the fact that all of them performs a preliminary edge detection step, which in turn applies a convolution kernel. Thus is somewhat expected that an additional blur does not affect severely the marker localization. Finally, it is interesting to note that higher angles (with respect to the marker's normal) lead to an higher accuracy (as long as the markers are still recognizable). This is explained by observing that the constraint of the reprojection increases with the angle of view. Overall, these experiments confirm that Rune-Tag always outperforms the other two tested techniques by about one order of magnitude. In practical terms the improvement is not negligible, in fact an error as low as 10^{-3} radians still produces a jitter of 1 millimeter when projected over a distance of 1 meter. While this is a reasonable performance for augmented reality applications, it can be unacceptable for obtaining precise contactless measures.

3.2 RUNE Tags for camera calibration

Since RUNE-129 provides an extremely robust yet precise way to localize many circular features we tried to use the proposed markers as a calibration target. In most cases, camera calibration is performed by exposing a well known pattern to the camera in many

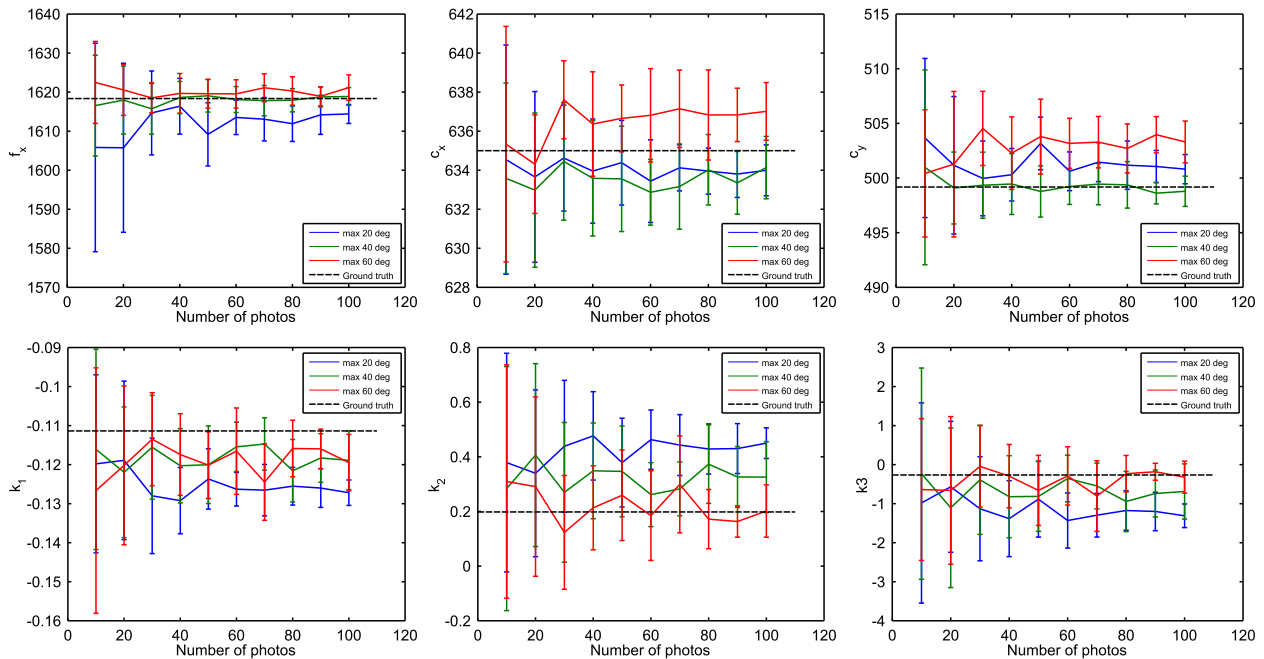


Fig. 7: Accuracy of camera calibration when using a single RUNE-129 as a dot-based calibration target. Camera poses has been divided into 3 groups based on the maximum angle between the camera z -axis and the marker plane. A random subset of photos is used to test the calibration varying the number of target exposures. In all the experiments we achieve a good accuracy with a decreasing st.dev. when increasing the number of photos.

different point of views. This allows the gathering of many 3D-2D point correspondences used to simultaneously recover the target pose, camera intrinsic parameters, and the lens distortion. Most of the time, a chessboard pattern is used since it provides a good set of feature points in the form of image corners. However, a manual chessboard boundary identification process is required for the limited robustness of such patterns against occlusions or illumination gradients. As a consequence, our fiducial markers may provide a very interesting alternative when an automatic calibration procedure is sought or an optimal visibility of the target cannot be guaranteed.

In Fig.7 we show the calibration results while calibrating a camera using a single RUNE-129 as calibration target and by varying the number of exposures used for each calibration. Specifically, we divided the camera poses (as given by PnP) into 3 major groups with respect to the angle between the camera z -axis and the marker plane normal. For each group, more than 200 photos are taken and a random subset of them are selected for each test to compose the plot. The ground truth is provided by a calibration performed with a 20×20 chessboard target exposed in 200 different poses using the method described in [32] to limit the errors due to printing misalignments. Camera calibration is performed by using the common technique described in [33] implemented by the OpenCV library [34].

Some interesting facts can be observed. First, the standard deviation of all the estimated parameters decrease by increasing the number of photos. This is

an expected behaviour that agrees with the accuracy experiments presented in section 3.1. Indeed, the more the number of target feature points given, the more the calibration error can be reduced by the non-linear optimization process. Second, the focal length estimation tends to be more accurate while considering the target poses spanning trough a greater range of angles (i.e. between 0 and 60 degrees). Differently, principal point seems to behave in the opposite way, giving better results when keeping the target plane more parallel to the image plane. This is probably due to the well known localization bias of the ellipse centers [35]. Third, the first two radial distortion parameters (i.e. k_1 and k_2) behave respectively like the principal point and the focal length. It has to be noted that a precise localization of the ellipse centers is only achievable in absence of distortion since the projective invariance of conics holds only for pure central projections. Therefore, we think that the calibration performance can be improved by estimating an initial calibration assuming no radial distortion followed by an iterative undistortion and re-localization of the circular features and a re-calibration of the camera intrinsics. Finally, we obtained no completely wrong calibrations due to mis-detections of the target thanks to the extremely resilient coding scheme used for markers identification.

In Fig.8 we compared the recovered camera intrinsic parameters varying the number of shots for RuneTag and a standard 10×10 Chessboard calibration target, calibrated with the method described in [33]. Both the two approaches shows comparable re-

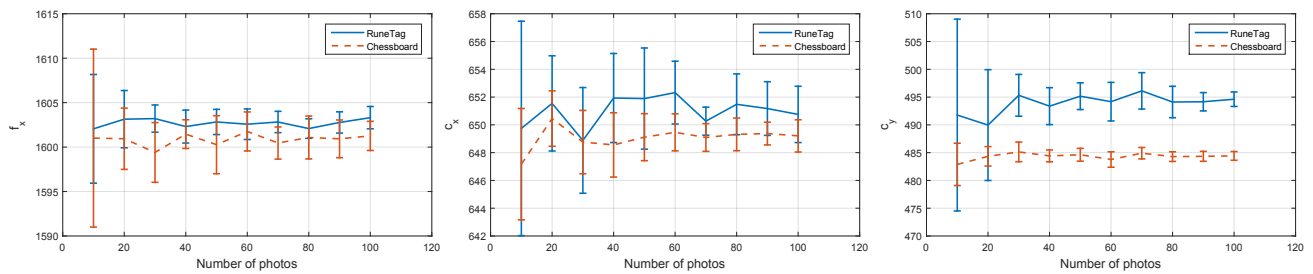


Fig. 8: Comparison between a calibration performed with RUNETag and chessboard target

sults, with the standard deviation decreasing while increasing the number of target exposures. Overall, chessboard target provides more stable results for the principal point while RuneTag target performs better in the focal length estimation.

3.3 Mono vs. Stereo Pose Estimation

To further test the camera pose estimation accuracy we compared the results achievable comparing a single camera setup (using PnP algorithm) with a calibrated stereo setup that can recover the pose by means of a 3D reconstruction of the marker followed by an estimation of the rigid motion with respect to the known model.

We started by calibrating a stereo rig composed by two 1.3 Mega-pixel cameras using a marker-based target as described in section 3.2. Then, we firmly positioned two RUNE-129 tags (with a diameter of 30 mm) at about 50 mm apart to a rigid metal rod so that they could only be moved without changing their relative position. The marker pair was then moved in a space varying from 230 to 320 mm in front of the stereo rig.

In the first experiment (Fig. 9, Left) we plotted the unknown distance between the two markers as estimated only by the first camera (in red), by the second (in green) and by using stereo reconstruction (blue) as a function of the angle between the first marker plane normal and the first camera optical axis. It can be

noted that the stereo case exhibit lower variance with respect to single-camera scenarios with some sparse outliers happening when the entire marker is not visible by both the cameras. Moreover, the distance measured by the mono case tends to be a little lower than the stereo one if the angle is below 30 degrees while increasing significantly for higher angles. This behaviour is probably due to the PnP algorithm that suffers for a non-isotropic error with respect to the three camera axis (i.e. the localization error on the camera z -axis is higher than the other two).

In (Fig. 9, Right) we compared the angle around the plane normal of a single RUNE-129 tag for mono (using the first camera) versus the stereo case. Ideally, the ratio between the two measures should be exactly 1 and so all the points should be disposed on the 45 degrees red line shown in the plot. We can observe that most of the measures are equally distributed above and below such line indicating no significant bias. This behaviour is consistent for all the angles spanning between 10 and 60 degrees since the overall geometrical shape of all the dots (i.e. minor and major axis length) remains constant if a rotation around the marker plane normal is applied. This suggests that the proposed tags may be used as a coarse registration initialization for a 3D scanner turntable.

We tested this possible real-world scenario to compare our proposed tags against the recently developed ArUco Tags [36] which exhibit similar reliability under

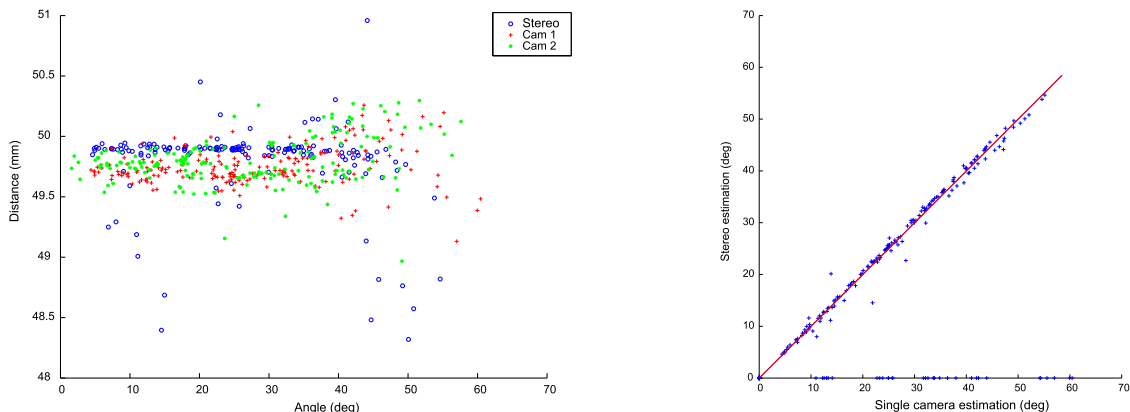


Fig. 9: Comparison between the pose accuracy for a single or stereo camera setup. Left: distance between two jointly moving markers as a function of the angle with respect to the first camera. Right: Angle around the marker plane normal as estimated by the first camera versus the stereo setup. Ideally, all the measures should lie on the 45 degrees red line.

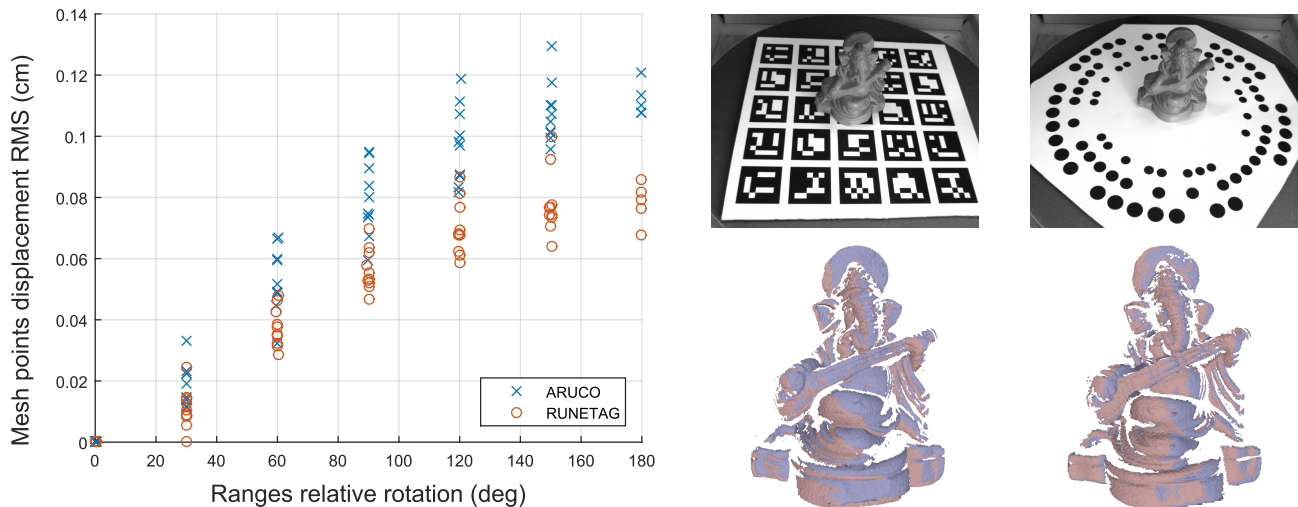


Fig. 10: Real-world reconstruction scenario. ArUco and RUNETag were placed on top of a scanner turntable to provide the initial coarse estimation for rangemap alignment. See text for details.

heavy occlusions. In (Fig. 10, Top-Right) we show two pictures of our setup, with a test object placed on top of one of the two kind of tags covering the entire turntable. To test the rangemap registration accuracy, we took 10 ranges spanning a complete 360 turn of the turntable. Then, for each couple of ranges, pose recovered with each tag was used to provide an initial coarse alignment which has been further refined with ICP. In (Fig. 10, Left) we plotted the distance between the initial coarse and the refined configuration of each range couple, in terms of root mean square distance of corresponding rangemap points. The rationale is that the worse the pose provided by the tag, the more ICP has to move each range for the fine alignment. In the plot we can clearly see how RuneTag provides lower RMS (so less ICP displacement was necessary) at any initial relative rotation. For both the tags, the initial relative rotation proportionally affects the pose estimation error. Finally, in (Fig. 10, Bottom-Right) we show a qualitative example of the coarse estimation provided by ArUco (Left) and RuneTag (Right).

3.4 Resilience to Occlusion and Illumination

One of the main characteristics of Rune-Tag is that it is very robust to occlusion. In section 2.3 we observed that RUNE-129 can be used to distinguish between

Occlusion	0%	10%	20%	50%	70%
RUNE-43	100%	69%	40%	0%	0%
RUNE-129	100%	100%	100%	100%	67%

TABLE 2: Recognition rate of the two proposed marker configurations with respect to the percentage of area occluded.

about 20.000 different tags and still be robust to occlusions as large as about 67% of the dots. By choosing different cyclic coding schemes is even possible to push this robustness even further, at the price of a lower number of available tags. In the first column of Fig. 11 we show how occlusion affects the accuracy of the pose estimation (i.e. how well the pose is estimated with fewer dots regardless to the ability of recognize the marker). Albeit a linear decreasing of the accuracy with respect to the occlusion can be observed, the precision is still quite reasonable also when most of the dots are not visible.

In Table 2 we show the recognition rate of the two proposed designs with respect to the percentage of marker area occluded. In the second column of Fig. 11 the robustness to illumination gradient is examined. The gradient itself is measured unit per pixel (i.e. quantity to add to each pixel value for a each pixel of distance from the image center). Overall, the proposed methods are not affected very much by the

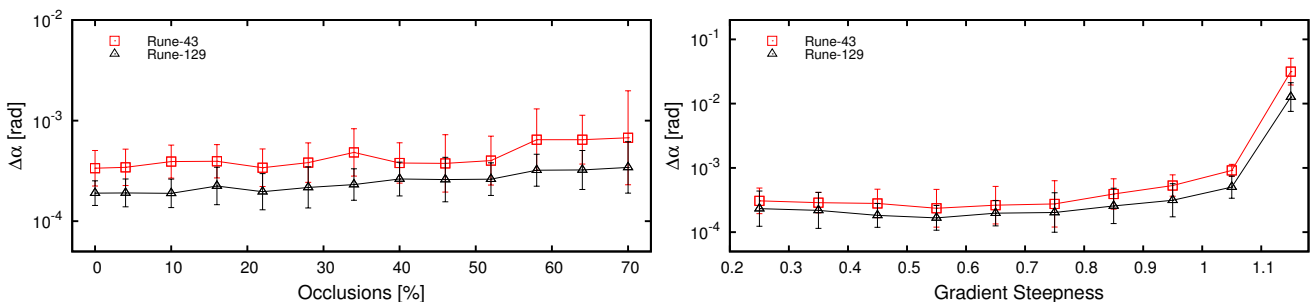


Fig. 11: Evaluation of the accuracy in the camera pose estimation of Rune-Tag with respect to occlusion (left column) and illumination gradient (right column).

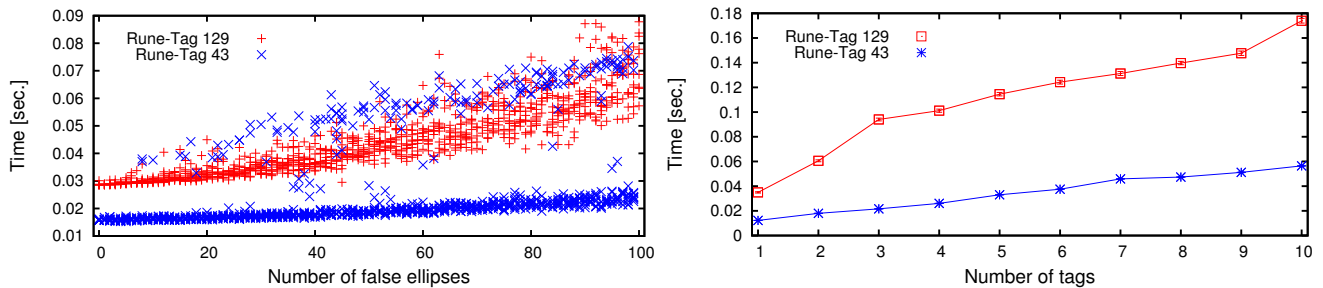


Fig. 12: Evaluation of the recognition time respectively when adding artificial false ellipses in the scene (left column) and with several markers (right column).

illumination gradient and break only when it become very large (in our setup an illumination gradient of 1 implies that pixels are completely saturated at 255 pixels from the image center). This agrees with the fact that a precise sub-pixel ellipse contour estimation is quite robust to steep changes in scene illumination.

3.5 Performance Evaluation

Our tag system is designed for improved accuracy and robustness rather than for high detection speed. This is quite apparent in Fig. 12, where we can see that the recognition could require from a minimum of about 15 ms (RUNE-43 with one tag and no noise) to a maximum of about 180 ms (RUNE-129 with 10 tags) with a consumer Core2 Duo PC, 2Ghz clock. By comparison ARToolkitPlus is about an order of magnitude faster [22]. However, it should be noted that, despite being slower, the frame rates reachable by Rune-Tag (from 60 to about 10 fps) can still be deemed as usable even for real-time applications (in particular when few markers are viewed at the same time).

3.6 Shortcomings and Limitations

In Fig. 13 some experiments with common occlusion scenarios are presented. In the first two shots an object is placed inside a RUNE-43 marker in a typical setup used for image-based shape reconstruction. In the following two frames a RUNE-129 marker is tested for its robustness to moderate and severe occlusion. At last, in Fig. 14 an inherent shortcoming of our design is highlighted. The high density exhibited by the more

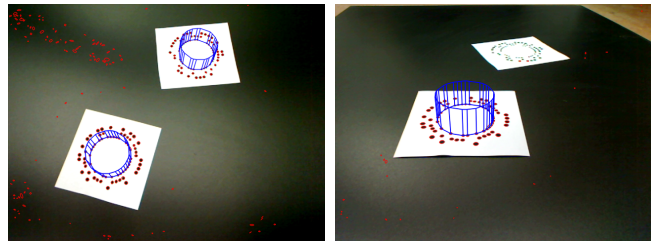


Fig. 14: Recognition fails when the marker is angled and far away from the camera and the ellipses blends together.

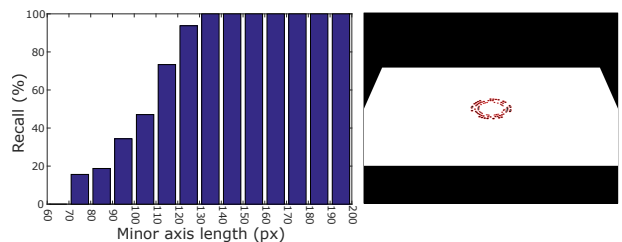


Fig. 15: Left: Identification percentage as a function of observed marker size. Right: Example of a successful identification of a far-away marker in the generated set.

packed markers may result in a failure of the ellipse detector whereas the tag is far away from the camera or very angled, causing the dots to become too small or blended.

To assess how much the dot size can be reduced while still allowing a reliable detection of the tags we generated a synthetic set of 471 images with a resolution 1024x768 pixels for different poses of a single RUNE129 tag. In Fig. 15 we plotted the percentage of the images for which the detection was

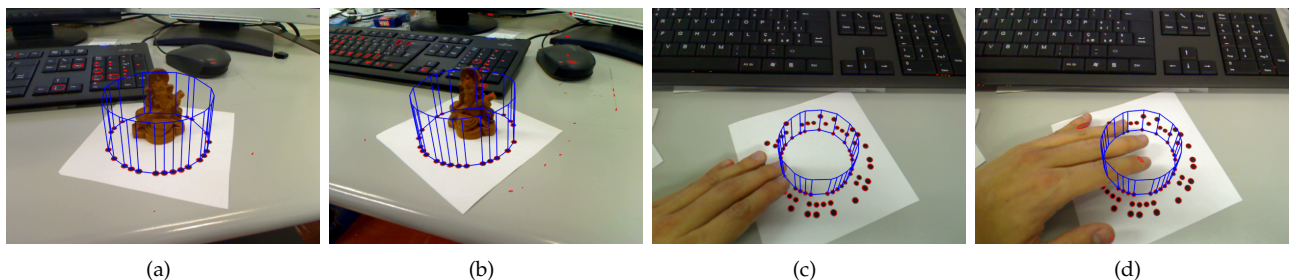


Fig. 13: Some examples of behaviour in real videos with occlusion. In (a) and (b) an object is placed inside the marker and the setup is rotated. In (c) and (d) the pose is recovered after medium and severe occlusion.

successful as a function of the minor axis length of the tag projection. Since the ratio between the innermost layer dot size and the whole marker size is 0.043, we see that with a minimum observed tag size of 130 px (internal dots of $\approx 6 \times 6$ px) we achieve almost a 100% detection rate which rapidly drops if the tag gets smaller. This suggests that the identification can still be reliable with extremely small dots size as soon as the ellipse detector is able to discriminate among each individual dot.

4 CONCLUSIONS

In this paper we proposed a novel fiducial marker system which heavily relies on the robust framework of cyclic codes to offer superior occlusion resilience, accurate detection and robustness against various types of noise. We improved on the seminal version proposed in [37] by investigating their usage even for the uncalibrated case and developed a fast technique to directly decode the symbol sequence encoded in each tag (i.e. a closed-form solution to recover the equivalence class and align the corresponding code).

Moreover, we expanded the experimental evaluation by testing its adequacy to be used as a typical dot-based camera calibration target. This is supported by also experiments to evaluate its behaviour comparing the pose estimation with both the mono and the stereo scenario.

Overall, RUNE-Tags offer many advantages over the existing fiducial marker systems. First, it gives the user the flexibility to trade-off the occlusion resilience with respect to the number of simultaneously detectable tags in a scene. Consequently, if one favors robustness against diversity, a very limited set of tags can be generated with high Hamming distance to guarantee extremely high error recovery rates. Second, the design itself may vary in the number of possible layers. The proposed single-layered RUNE-43 exhibit limited occlusion resilience while offering plenty of space in the marker interior for any additional payload or even for placing a physical object for reconstruction task. Third, by providing many circular features on a single tag we not only achieve an order of magnitude better pose recovery with respect to other tags but we managed to use the tag itself as a calibration target.

Finally, while slower than other techniques, our method is fast enough to be used in real-time applications. However, the severe packing of circular points may cause the ellipse detector to fail especially when dealing with low resolution, high angles or motion-blurred images. This limitation can be easily relieved by using a simpler marker, such as RUNE-43, which allows for a more extended range while still providing a satisfactory precision.

An implementation of our proposed tags is available at <http://www.dsi.unive.it/~bergamasco/runetag/>

REFERENCES

- [1] G. Thomas, "Derivation of studio camera position and motion from the camera image," Feb. 10 2004, uS Patent RE38,420. [Online]. Available: <http://www.google.com.ar/patents/USRE38420>
- [2] A. Herout, I. Szentandrasei, M. Zacharia, M. Dubska, and R. Kajan, "Five shades of grey for fast and reliable camera pose estimation," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 1384–1390.
- [3] N. Hagbi, O. Bergig, J. El-Sana, and M. Billinghurst, "Shape Recognition and Pose Estimation for Mobile Augmented Reality," in *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*. IEEE Computer Press, 2009, pp. 65–71.
- [4] S. Daftry, M. Maurer, A. Wendel, and H. Bischof, "Flexible and user-centric camera calibration using planar fiducial markers," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2013.
- [5] I. Szentandrasei, M. Zacharias, J. Havel, A. Herout, M. Dubska, and R. Kajan, "Uniform marker fields: Camera localization by orientable de bruijn tori," in *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, Nov 2012, pp. 319–320.
- [6] L. Gatrell, W. Hoff, and C. Sklair, "Robust image features: Concentric contrasting circles and their image extraction," in *Proc. of Cooperative Intelligent Robotics in Space*. Washington, USA: SPIE, 1991.
- [7] Y. Cho, J. Lee, and U. Neumann, "A multi-ring color fiducial system and a rule-based detection method for scalable fiducial-tracking augmented reality," in *Proceedings of International Workshop on Augmented Reality*, 1998.
- [8] V. A. Knyaz and R. V. Sibiriyakov, "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*, 1998.
- [9] 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*, ser. ISMAR '02. Washington, DC, USA: IEEE Computer Society, 2002.
- [10] D. Claus and A. W. Fitzgibbon, "Reliable automatic calibration of a marker-based position tracking system," in *IEEE Workshop on Applications of Computer Vision*, 2005.
- [11] J. Rekimoto and Y. Ayatsuka, "CyberCode: designing augmented reality environments with visual tags," in *DARE '00: Proceedings of DARE 2000 on Designing augmented reality environments*. New York, NY, USA: ACM, 2000.
- [12] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005.
- [13] J. Mooser, S. You, and U. Neumann, "Tricodes: A barcode-like fiducial design for augmented reality media," *Multimedia and Expo, IEEE International Conference on*, 2006.
- [14] L. Teixeira, M. Loaiza, A. Raposo, and M. Gattass, "Augmented reality using projective invariant patterns," in *Advances in Visual Computing*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5358.
- [15] V. S. Tsonisp, K. V. Ch, and P. E. Trahaniaslj, "Landmark-based navigation using projective invariants," in *Proceedings of the 1998 IEEE Intl. Conf. on Intelligent Robots and Systems*. Victoria, Canada: IEEE Computer Society, 1998.
- [16] R. V. Liere and J. D. Mulder, "Optical tracking using projective invariant marker pattern properties," in *Proceedings of the IEEE Virtual Reality Conference*. IEEE Press, 2003.
- [17] A. van Rhijn and J. D. Mulder, "Optical tracking using line pencil fiducials," in *Proceedings of the eurographics symposium on virtual environments*, 2004.
- [18] D. Q. Huynh, "The cross ratio: A revisit to its probability density function," in *Proceedings of the British Machine Vision Conference BMVC 2000*, 2000.
- [19] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," in *Proceedings of the 2nd IEEE and ACM International*

Workshop on Augmented Reality. Washington, DC, USA: IEEE Computer Society, 1999.

- [20] M. Maidi, J.-Y. Didier, F. Ababsa, and M. Mallem, "A performance study for camera pose estimation using visual marker based tracking," *Mach. Vision Appl.*, vol. 21, 2010.
- [21] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 32, no. 7, 2010.
- [22] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real time detection and tracking for augmented reality on mobile phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 99, 2010.
- [23] A. Stathakis and E. Petriu, "Robust pseudo-random fiducial marker for indoor localization," in *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, Sept 2011, pp. 19–24.
- [24] A. W. Fitzgibbon and R. B. Fisher, "A buyer's guide to conic fitting," in *Proceedings of the 6th British Conference on Machine Vision (Vol. 2)*, ser. BMVC '95. Surrey, UK, UK: BMVA Press, 1995, pp. 513–522. [Online]. Available: <http://dl.acm.org/citation.cfm?id=243124.243148>
- [25] J. Ouellet and P. Hebert, "Precise ellipse estimation without contour point extraction," *Mach. Vision Appl.*, vol. 21, 2009.
- [26] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "A robust and accumulator-free ellipse hough transform," in *Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM, 2004.
- [27] Q. Chen, H. Wu, and T. Wada, "Camera calibration with two arbitrary coplanar circles," in *European Conference on Computer Vision - ECCV*, 2004.
- [28] F. Bergamasco, L. Cosmo, A. Albarelli, and A. Torsello, "Camera calibration from coplanar circles," in *22nd International Conference on Pattern Recognition (ICPR 2014)*, Aug 2014.
- [29] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, 2nd ed. North-holland Publishing Company, 1978.
- [30] E. Orsini and M. Sala, "Correcting errors and erasures via the syndrome variety," *Journal of Pure and Applied Algebra*, vol. 200, no. 1–2, pp. 191–226, 2005.
- [31] G. Forney, "On decoding bch codes," *Information Theory, IEEE Transactions on*, vol. 11, no. 4, pp. 549–557, Oct 1965.
- [32] A. Albarelli, E. Rodolà, and A. Torsello, "Robust camera calibration using inaccurate targets," in *Proc. BMVC*, 2010, pp. 16.1–10, doi:10.5244/C.24.16.
- [33] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, Nov 2000.
- [34] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, 1st ed. O'Reilly Media, Inc., 2008.
- [35] J. Mallon and P. F. Whelan, "Which pattern? biasing aspects of planar calibration patterns and detection methods," *Pattern Recogn. Lett.*, vol. 28, no. 8, pp. 921–930, Jun. 2007.
- [36] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marn-Jimnez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280 – 2292, 2014.
- [37] F. Bergamasco, A. Albarelli, E. Rodolà, and A. Torsello, "Rune-tag: A high accuracy fiducial marker with strong occlusion resilience," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 113–120.



Filippo Bergamasco received his Ph.D. in Computer Science in from the University Ca' Foscari Venice, Italy, in 2015 and is currently a post-doc researcher at Ca'Foscari University of Venice in joint collaboration with CNR-ISMAR to develop a vision-based 3D sea surface reconstruction pipeline from moving vessels. His research interests are mainly in the area of computer vision, spreading from 3D reconstruction, camera calibration to structure from motion and augmented reality. He is currently interested in high-precision photogrammetry, innovative non-parametric models for camera calibration, ego-motion estimation and 3D reconstruction applied to moving/nonrigid objects.



Andrea Albarelli worked in the industry for 10 years before receiving his Ph.D. in Computer Science in from the University Ca' Foscari Venice in 2010. He has taught Computer Vision, Computer Architecture and Information Theory. He co-authored 40 peer-reviewer technical papers, mainly in the field of Computer Vision, with particular attention to issues of representation and processing of 3D data and the adoption of Game Theory in the context of Pattern Recognition problems, ranging from point-pattern matching to surface registration. He has participated in several research and technology transfer projects funded by both public and private partners. Since 2010, he is a founding member of an academic spin-off acting as a bridge between industry and Computer Vision research. In 2010 he was winner of the tender "IMPRESA", sponsored by the Ministry of Economic Development, the prize "Working Capital" for innovative young researchers and the "NVIDIA Best Paper Award". In 2011 he won the "Award for Research" established by Venice University to reward young researchers exhibiting the highest scientific impact.



Luca Cosmo received a MSc degree with honors in Computer Science from the University of Venice, Italy, in 2012. He is currently a PhD candidate at the University of Venice where his field of research is the Non-Rigid matching. He has been involved in computer vision projects for industry and museum exhibitions. For hobby he deals with the development of the graphical engine and networking of PC strategy games.



Emanuele Rodolà received the BS/MS degree in computer engineering from the University of Rome Tor Vergata, Italy, in 2008 and the PhD degree in computer science from Ca' Foscari University of Venice in 2012. In 2012 he was a JSPS Research Fellow at the University of Tokyo, Intelligent Systems and Informatics Lab, and since 2013 he is an Alexander von Humboldt Postdoctoral Fellow at the Technical University of Munich, Computer Vision group. He has been in the program committees of various international conferences and journals, and was recognized as Outstanding Reviewer at CVPR 2013, ECCV 2014 and CVPR 2015. His research interests include 3D shape analysis, matching, reconstruction and has around 40 publications on these topics in refereed conferences and journals.



Andrea Torsello received his PhD in computer science at the University of York, UK, and is currently an assistant professor at Ca' Foscari University of Venice, Italy. His research interests are in the areas of computer vision and pattern recognition, in particular, the interplay between stochastic and structural approaches as well as game-theoretic models. Dr. Torsello has published over 100 technical papers in refereed journals and conference proceedings, has been in the program committees of various international conferences and workshops and is currently editor for the *Pattern Recognition* and *Pattern Recognition Letters* journals.