RUNE-Tag: a High Accuracy Fiducial Marker with Strong Occlusion Resilience

Filippo Bergamasco, Andrea Albarelli, Emanuele Rodolà and Andrea Torsello
Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia - via Torino, 155 - 30172 Venice Italy
`http://www.dais.unive.it`

Abstract

Over the last decades fiducial markers have provided widely adopted tools to add reliable model-based features into an otherwise general scene. Given their central role in many computer vision tasks, countless different solutions have been proposed in the literature. Some designs are focused on the accuracy of the recovered camera pose with respect to the tag; some other concentrate on reaching high detection speed or on recognizing a large number of distinct markers in the scene. In such a crowded area both the researcher and the practitioner are licensed to wonder if there is any need to introduce yet another approach. Nevertheless, with this paper, we would like to present a general purpose fiducial marker system that can be deemed to add some valuable features to the pack. Specifically, by exploiting the projective properties of a circular set of sizeable dots, we propose a detection algorithm that is highly accurate. Further, applying a dot pattern scheme derived from error-correcting codes, allows for robustness with respect to very large occlusions. In addition, the design of the marker itself is flexible enough to accommodate different requirements in terms of pose accuracy and number of patterns. The overall performance of the marker system is evaluated in an extensive experimental section, where a comparison with a well-known baseline technique is presented.

1. Introduction

A fiducial marker is, in its broadest definition, any artificial object consistent with a known model that is placed in a scene. At the current state-of-the-art such artifacts are still only choice whenever a high level of precision and repeatability in image-based measurement is required. This is, for instance, the case with accurate camera pose estimation, 3D structure-from-motion or, more in general, any flavor of vision-driven dimensional assessment task. Of course a deluge of approaches have been proposed in order to obtain a reasonable performance by relying only on natural features already present in the scene. To this ex-

tent, several repeatable and distinctive interest point detection and matching techniques have been proposed over the years. While in some scenarios such approaches can obtain satisfactory results, they still suffer from shortcomings that severely hinder their broader use. Specifically, the lack of a well known model limits their usefulness in pose estimation and, even when such a model can be inferred (for instance by using bundle adjustment) its accuracy heavily depends on the correctness of localization and matching. Moreover, the availability and quality of natural features in a scene is not guaranteed in general. Indeed, the surface smoothness found in most man-made objects can easily lead to scenes that are very poor in features. Finally, photometric inconsistencies due to reflective or translucent materials jeopardizes the repeatability of the detected points. For this reasons, it is not surprising that artificial fiducial tags continue to be widely used and are still an active research topic. For practical purposes, most markers are crafted in such a way as to be easily detected and recognized in images produced by a pinhole-modeled camera. In this sense, their design leverages the projective invariants that characterizes geometrical entities such as lines, planes and conics. It is reasonable 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 the point of view of the camera. This allows both for an easy detection and a quite straightforward rectification of the circle plane. In his seminal work Gatrell [7] proposes to use a set of highly contrasted concentric circles and to validate a candidate marker by exploiting the compatibility between the centroids of the ellipses found. By alternating white and black circles a few bits of information can be encoded in the marker itself. In [3] the concentric circle approach is enhanced by adding colors and multiple scales. In [11] and [16] dedicated "data rings" are added to the fiducial design. A set of four circles located at the corner of a square is adopted by [4]: in this case an identification pattern is placed at the centroid of the four dots in order to distinguish between different targets. This ability to recognize the viewed markers is very important

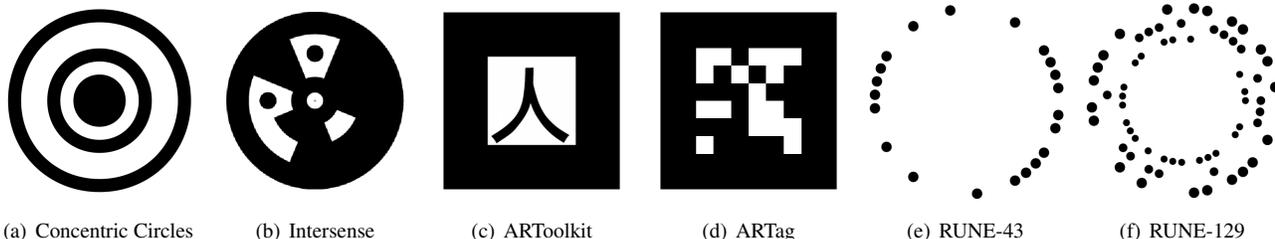| (a) Concentric Circles | (b) Intersense | (c) ARToolkit | (d) ARTag | (e) RUNE-43 | (f) RUNE-129 |

Figure 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 (a) and by the appearance of the sectors in (b). The black square border enables detection in (c) and (d), but while ARToolkit uses image correlation to differentiate markers, ARTag relies in error-correcting binary codes. Finally, in (e) and (f) we show two examples of RUNE-Tags.

for complex scenes where more than a single fiducial is required, furthermore, the availability of a coding scheme allows for an additional validation step and lowers the number of false positives. While coded patterns are widely used (see for instance [18, 5, 15]) it is interesting to note that many papers suggest the use of the cross ratio among detected points [19, 20, 12] or lines [21]. A clear 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. In fact the cross ratio is a single scalar with a strongly non-uniform distribution [8] and this limits the number of well-spaced different values that can possibly be generated. Also the projective invariance of lines is frequently used in the design of fiducial markers. Almost invariably this feature is exploited by detecting the border edges of a highly contrasted quadrilateral block. This happens, for instance, with the very well known ARToolkit [10] system which is freely available and adopted in countless virtual reality applications. Thanks to its easy detection and the high accuracy that can be obtained in pose recovery [14], this solution is retained in many recent approaches, such as ARTag [6] and ARToolkitPlus [22]. These two latter methods replace the recognition technique of ARToolkit, which is based on image correlation, with a binary coded pattern (see Fig. 1). The use of an error-correcting code makes the marker identification very robust, in fact we can deem these designs as the most successful from an applicative point of view.

In this paper we introduce a novel fiducial marker system that takes advantage of the same basic features for detection and recognition purposes. The marker is characterized by a circular arrangement dots at fixed angular positions in one or more concentric rings. Within this design, the projective properties of both the atomic dots and the rings they compose are exploited to make the processing fast and reliable. In the following section we describe the general nature of our marker, the algorithm proposed for its detection and the coding scheme to be used for robust recognition. In the experimental section we validate the proposed approach

by comparing its performance with two widely used marker systems and by testing its robustness under a wide range of noise sources.

2. Rings of Unconnected Ellipses

The proposed tag is built by partitioning a disc in several evenly distributed *sectors*. Each sector, in turn, can be divided into a number of concentric rings, which we call *levels*. Each pair made up of a sector and a level defines a *slot* where a *dot* can be placed. Finally, each dot is a circular feature whose radius is proportional to the radius of the level at which the dot is placed. Within this design the regular organization of the dots enables easy localization and, by properly populating each slot, it is possible to bind some information to the tag. In Fig. 1(e) a tag built with 43 sectors and just one level is shown. In Fig. 1(f) we add two more levels: note that the dot size decreases for the inner levels. We will explain in the following sections how this structure is also flexible and well suitable to deal with many scenarios.

2.1. Fast and Robust Detection in Projective Images

Both the dots and the ideal rings on which they are disposed appear as ellipses under general projective transform. Thus, the first step of the localization procedure is to try to locate the dots by finding all the ellipses in the scene. For this purpose we use the ellipse detector supplied by the OpenCV [1] library, but any other suitable technique would be fine. The dots candidates found at this stage can be considered the starting point for our algorithm. A common approach would consists in the use of a RANSAC scheme on features centers in order to locate the dots that belong to the

Total ellipses	10	50	100	500
Full (RANSAC)	252	2118760	75287520	$> 10^{10}$
Proposed method	45	1225	4950	124750

Figure 2. Number of maximum steps required for ellipse testing.

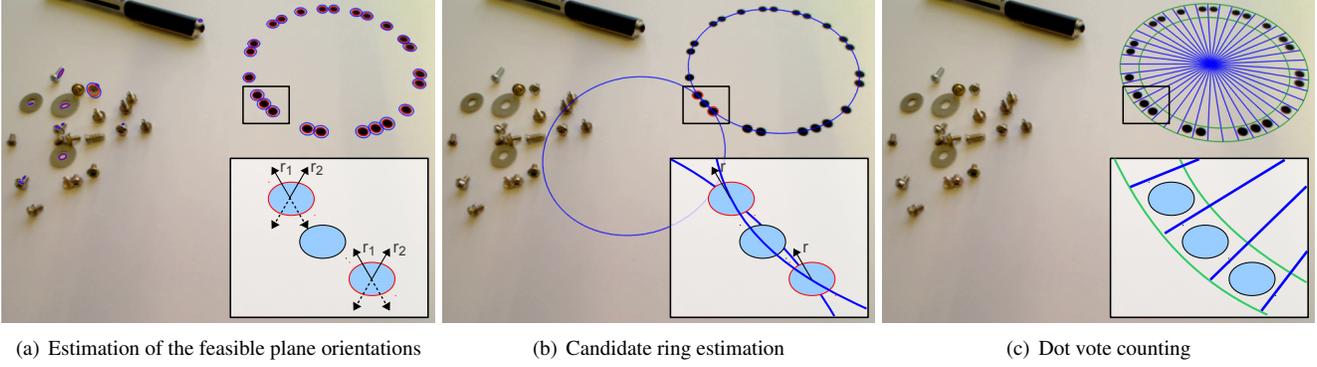(a) Estimation of the feasible plane orientations   (b) Candidate ring estimation   (c) Dot vote counting

Figure 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).

same marker (if any) and to separate them them from false positives. Unfortunately, five points are needed to characterize an ellipse, thus the use of RANSAC (especially with many false positives) could lead to an intractable problem (see Fig. 2). Since the marker itself can contain more than one hundred dots, it is obvious that this approach is not feasible. A possible alternative could be the use of some specialized Hough Transform [23], but also this solution would not work since the relatively low number of dots (coupled with the high dimensionality of the parameter space) hinders the ability of the bins to accumulate enough votes for a reliable detection. In order to cluster dots candidates into coherent rings we need to exploit some additional information. Specifically, after the initial ellipse detection the full conic associated to each dot candidate is known. While from this single conic it is not possible to recover the full the camera pose, nevertheless we can estimate a rotation that transform the ellipse into a circle. Following [2], the first step for recovering such rotation is to normalize the conic associated to the dot, obtaining:

$$\mathbf{Q} = \begin{bmatrix} A & B & -\frac{D}{f} \\ B & C & -\frac{E}{f} \\ -\frac{D}{f} & -\frac{E}{f} & -\frac{F}{f^2} \end{bmatrix}$$

where f is the focal length of the camera that captured the scene and $Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0$ is the implicit equation of the ellipse found. The $\mathbf{Q}$ is then decomposed via SVD:

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

The required rotation can thus be computed (up to some parameters) as:

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} gcos\alpha & s_1 gsin\alpha & s_2 h \\ sin\alpha & -s_1 cos\alpha & 0 \\ s_1 s_2 hcos\alpha & s_2 hsin\alpha & -s_1 g \end{bmatrix}$$

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

Here α is an arbitrary rotation around the normal of the marker plane. Since we are not interested in the complete pose (which is not even possible to recover) we can just fix such angle to 0 and obtain:

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} g & 0 & s_2 h \\ 0 & -s_1 & 0 \\ s_1 s_2 h & 0 & -s_1 g \end{bmatrix}$$

Finally s_1 and s_2 are two free signs, which leave us with four possible rotation matrices, defining four different orientations. Two of these orientation can be eliminated, as they are discording with the line of sight. The other two must be evaluated for each detected ellipse: we can define them as r_1 and r_2 (see Fig. 3(a)). At this point it is possible to search for whole markers. For all the pairs of detected ellipses the rotations are combined to form four feasible rotation pair. These are filtered eliminating the pairs with an inner product above a fixed threshold and then the best pair of rotation is selected by applying the average of the rotations (as quaternions) to both ellipses and a choosing the pair with the minimal distance between the mean radii of the rectified ellipses. The rationale of the filtering is to avoid to choose ellipses with discordant orientations (as the marker is planar) and the compatibility score takes in account that the dots on the same ring should be exactly the same size on the rectified plane. If a good average rotation r is found then exactly two hypothesis about the ring location can be made. In fact we know both the angle between camera and marker planes and the size of the dots on the rectified plane. Since the ratio between the radii of each level and the dots that it contains is known and constant (regardless of the level) we can estimate the radius of the ring. Finally we can fit such ring of know radius to the two dots examined and thus reproject on the image plane the two possible solutions (Fig. 3(b)). In this way we get
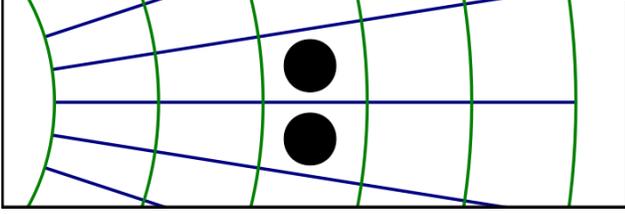
Figure 4. Detection grid for a Rune-Tag with multiple levels

two main advantages. The first one is at most $O(n^2)$ candidate rings have to be tested, were n is the number of the ellipses found (in Fig. 2 we can see that the problem becomes tractable). The second advantage is that, as opposed to many other approaches, the vote binning and the recovery of the code happens entirely in the image space, thus no picture rectification is required. Note that the counting of the dots happens by reprojecting the circular grid made by sectors and levels on the image (Fig. 3(c)). Of course if more than one ring is expected we need to project the additional levels both inward and outward (see Fig. 4). This is due to the fact that even if a correct ring is detected we still do not know at which level it is located since the ratio of the dots is scaled accordingly.

2.2. Marker Recognition and Coding Strategies

Once the candidate ellipses are found we are left with two coupled problems: the first is that of assigning correspondences between the candidates ellipses and the circles in the marker, or, equivalently, to find an alignment around the orthogonal axis of the marker; the second is that of recognizing which of several markers we are dealing with.

The problem is further complicated by the fact that misdetections and occlusions make the matching non exact. Here we chose to cast the problem into the solid and well-developed mathematical framework of coding theory where the circle pattern corresponds to a code with clearly designed properties and error-correcting capabilities. In what follows we will give a brief review of the theory needed to build and decode the markers. We refer to [13] for a more in-depth introduction to the field.

A *block code* of length n over a set of Symbols S is a set $C \subset S^n$ and the elements of a code are called codewords. The *Hamming distance* $dH : S \times S \to \mathbb{N}$ is the number of symbols that differ between two codeword, i.e.,

$$dH(u, v) = |i \text{ s.t. } u_i \neq v_i, i = 1...n|$$

The *Hamming distance* of a code is the minimum distance between all the codewords: $dH(C) = \min_{u,v \in C} dH(u, v)$. A code with Hamming distance d can detect $d - 1$ errors and correct $\lfloor (d - 1)/2 \rfloor$ or $d - 1$ erasures (i.e., situations in which we have unreadable rather than wrong symbols).

Let $q \in \mathbb{N}$ such that $q = p^k$, for prime a p and an integer $k \geq 1$, we denote with $\mathbb{F}_q$ the 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 of length n and dimension k has q^k distinct codewords and is subject to the singleton bound: $d \leq n-k+1$, thus, with a fixed code length n, higher error correcting capabilities are payed with a smaller number of available codewords.

In our setting we map the point patterns around the circle to a codeword, but, since on a circle we do not have a starting position of the code, we have to take into account all cyclic shifts of a pattern. A linear code C is called *cyclic* if any cyclic shift of a codeword is still a codeword, i.e.

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

There is a bijection between the vectors of $(\mathbb{F}_q)^n$ and residue class of $\mathbb{F}_q[x]$ modulo division by $x^n - 1$.

$$v = (v_0, \ldots, v_{n-1}) \Leftrightarrow v_0 + v_1 x + \cdots + v_{n-1} x^{n-1}.$$

Multiplying a polynomial form of a code by x modulo $x^n - 1$ corresponds to cyclic shift:

$$x(c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}) = c_{n-1} + c_0 x + \cdots + c_{n-2} x^{n-1}.$$

Further, C is a cyclic code if and only if C is an ideal of the quotient group of the polynomial ring $\mathbb{F}_q[x]$ modulo division by $x^n - 1$. This means that all cyclic codes in polynomial form are multiples of a monic *generator polynomial* $g(x)$ which divides $x^n - 1$ in $\mathbb{F}_q$. Thus, if $g(x)$ is a generator polynomial of degree m, all codewords can be obtained by any mapping a polynomial $p(x) \in \mathbb{F}_q[x]$ of degree at most $n - m - 1m$ into $p(x)g(x) \mod x^n - 1$.

Using a cyclic code of distance $2e+1$ guarantees that we can correct e misdetections, regardless of the actual alignment of the patterns. Moreover, we can decode the marker used and recover the alignment at the same time. Since all the cyclic shifts are codes, we can group the codewords into *cyclic equivalence classes*, such that two codewords are in the same class if one can be obtained from the other with a cyclic shift. Clearly, the number of elements in a cyclic equivalence class divides n, so by choosing n prime, we only have classes where all the codewords are distinct, or classes composed of one element, i.e., constant codewords with n repetitions of the same symbol. The latter group is composed of which are at most q codewords and can be easily eliminated. In our marker setting, the choice of the marker is encoded by the cyclic equivalence class, while the actual alignment of the circles can be obtained from the detected element within the class.

In this paper we are restricting our analysis to the correction of random errors or erasures, but it is worth noting that cyclic codes have been used to detect and correct burst errors, i.e. errors that are spatially coherent, like we have in the case of occlusions.
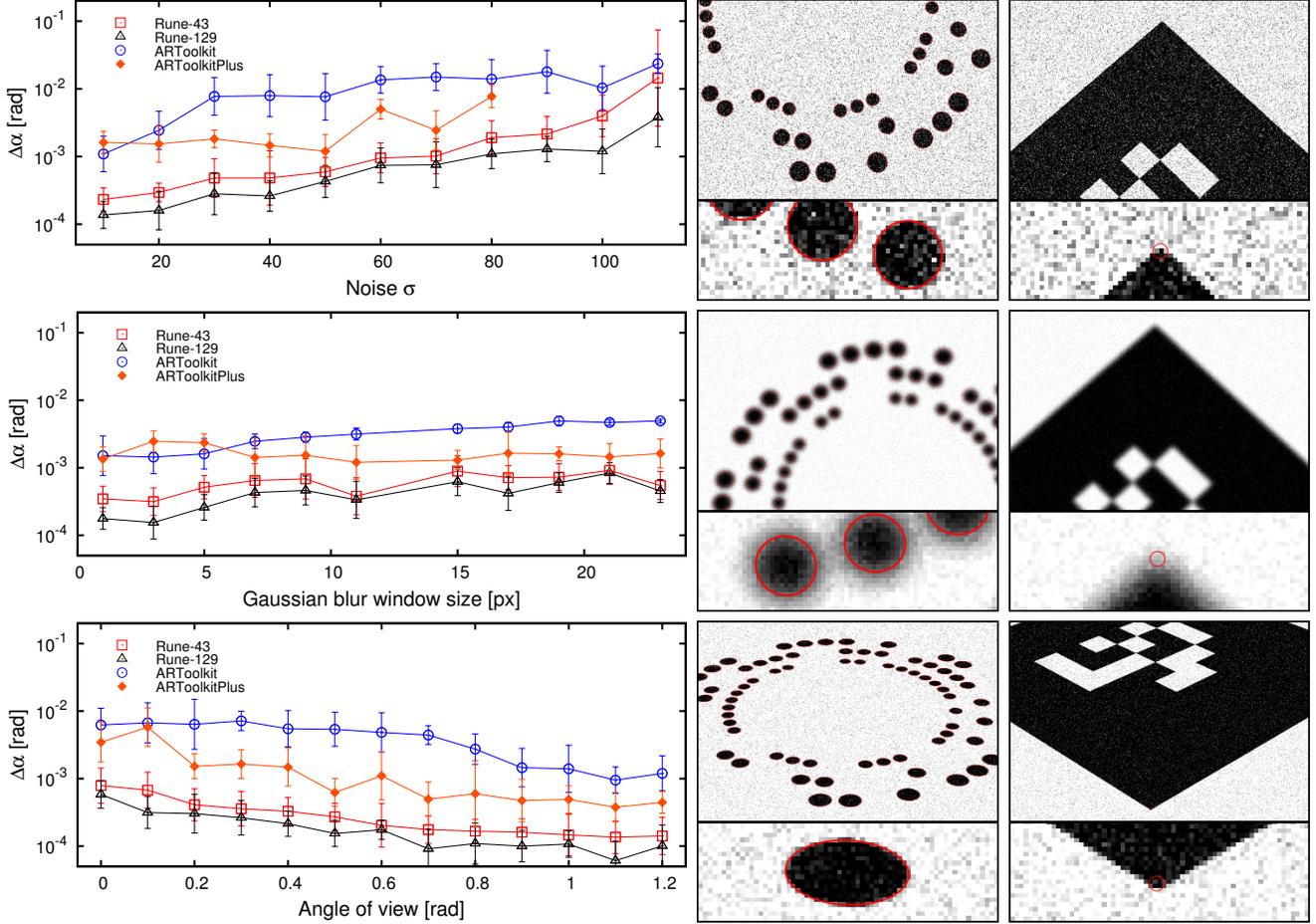
Figure 5. Evaluation of the accuracy in the camera pose estimation with respect to different scene conditions. Examples of the detected features are shown for RUNE-129 (first image column) and ARToolkitPlus (second image column).

Specifically, we experiment with two distinct codes. The first code (RUNE-43) is formed of a single circular pattern of circles that can be present or absent in 43 different angular slots. In this situation we encode the pattern as a vector in $(\mathbb{Z}_2)^{43}$, where $\mathbb{Z}_2$ is the remainder class modulo 2. For this code we chose the generator polynomial

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

which provides a cyclic code of dimension 15 giving 762 different markers (equivalence classes) with a minimum distance of 13, allowing us to correct up to 6 errors.

The second code (RUNE-129) is formed of a three concentric pattern of circles in 43 different angular slots. In this situation we have 8 possible patterns for each angular slot. We hold out the pattern with no circles to detect erasures due to occlusions and we encode the remaining 7 as a vector in $(\mathbb{Z}_7)^{43}$. For this code we chose the generator

polynomial

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

providing a cyclic code of dimension 7 which gives 19152 different markers with a minimum distance of 30, allowing us to correct up to 14 errors, or 29 erasures, or any combination of e errors and c erasures such that $2e + c \leq 29$. For efficient algorithms to decode the patterns and correct the error we refer to the literature [13].

2.3. Estimation of the Camera Pose

By using the detected and labelled ellipses it is now possible to estimate the camera pose. Since the geometry of the original marker is known any algorithm that solves the PnP problem can be used. In our test we used the *solvePnP* function available from OpenCV. However it should be noted
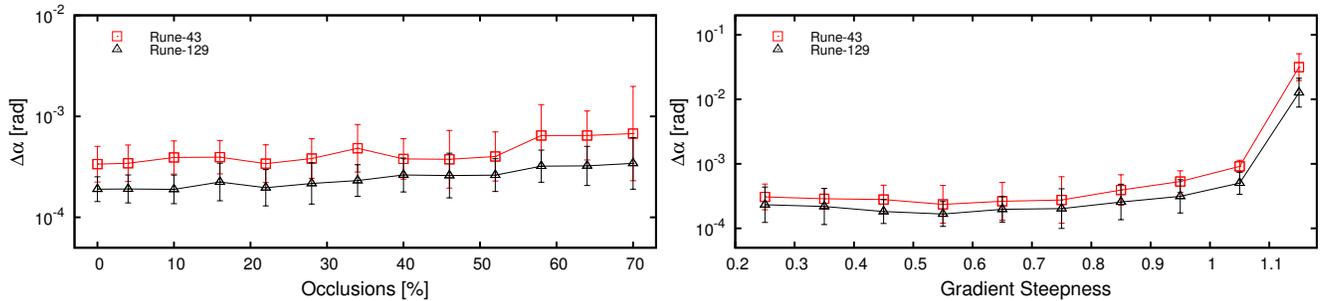
Figure 6. Evaluation of the accuracy in the camera pose estimation of RUNE-Tag with respect to occlusion (left column) and illumination gradient (right column).
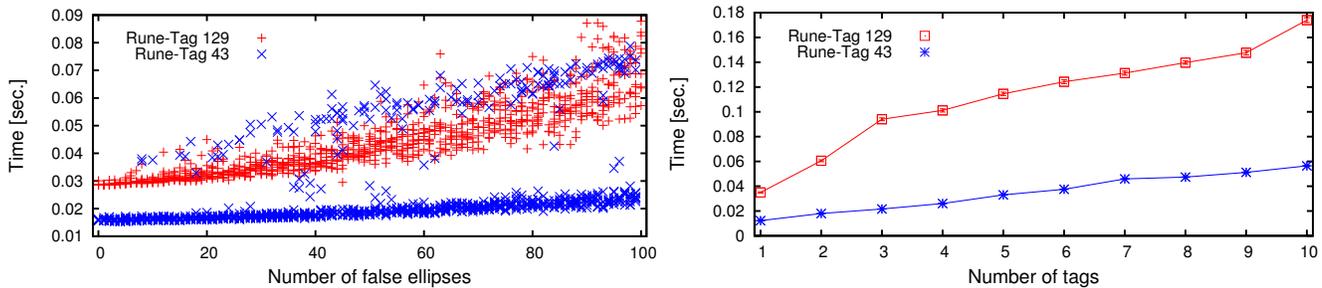

Figure 7. Evaluation of the recognition time respectively when adding artificial false ellipses in the scene (left column) and with several markers (right column).

that, while the estimated ellipse centers can be good enough for the detection step, it could be reasonable to refine them in order to recover a more accurate pose. Since this is done only when a marker is found and recognized we can indulge and dedicate a little more computational resources at this stage. In this paper we used the robust ellipse refinement presented in [17]. In addition to a more accurate localization it could be useful to correct also the projective displacement of the ellipses centers. However, according to our tests, such correction gives in general no advantage and sometimes leads to slightly less accurate results. Finally we also tried the direct method outlined in [9], but we obtained very unstable results, especially with small and skewed ellipses.

3. Experimental Validation

In this section the accuracy and speed of the Rune-Tag fiducial markers is evaluated and compared with the results obtained by ARToolkit and ARToolkitPlus. Both tags with one level (RUNE-43) and three levels (RUNE-129) are tested. All the experiments have been performed on typical a desktop PC equipped with a 1.6Ghz Intel Core Duo processor. The accuracy of the recovered pose is measured as the angular difference between the ground truth camera orientation and the pose obtained. Such ground truth is known since the test images are synthetically generated under different condition of nose, illumination, viewing direction, etc. The implementations of ARToolkit and ARToolkitPlus

used are the ones freely available at the respective websites. The real images are taken with a 640x480 CMOS webcam.

3.1. Accuracy and Baseline Comparisons

In Fig. 5 the accuracy of our markers is evaluated. In the first test an additive Gaussian noise was added to images with an average view angle of 0.3 radians and no artificial blur added. The performance of all methods get worse with increasing levels of noise and ARToolkitPlus, while in general more accurate than ARToolkit, 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 oblique angles 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
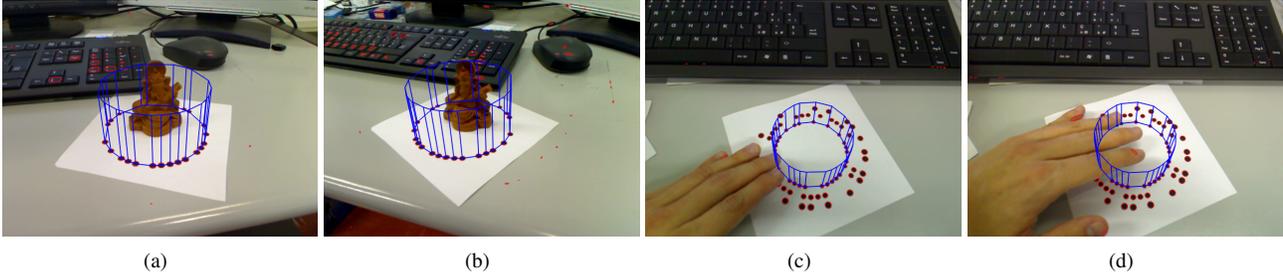
Figure 8. Some examples of behaviour in real videos with occlusion. In (a) and (b) an object is placed inside the marker and the setup is rotated. In (c) and (d) the pose is recovered after medium and severe occlusion.

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. Resilience to Occlusion and Illumination

One of the main characteristics of Rune-Tag is that it is very robust to occlusion. In section 2.2 we observed that RUNE-129 can be used to distinguish between about 20.000 different tags and still be robust to occlusions as large as about 2/3 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. 6 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 observer, the precision is still quite reasonable also when most of the dots are not visible. In Fig. 9 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. 6 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 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).

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

Figure 9. Recognition rate of the two proposed marker configurations with respect to the percentage of area occluded.

3.3. Performance Evaluation

Our tag system is designed for improved accuracy and robustness rather than for high detection speed. This is quite apparent in Fig. 7, where we can see that the recognition could require from a minimum of about 15 ms (RUNE-43 with one tag an no noise) to a maximum of about 180 ms (RUNE-129 with 10 tags). By comparison ARToolkit-Plus 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).

3.4. Behaviour with Real Images

In addition to the evaluation with synthetic images we also performed some qualitative tests on real videos. In Fig. 8 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. 10 an inherent shortcoming of our design is highlighted. The high density exhibited by the more 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.
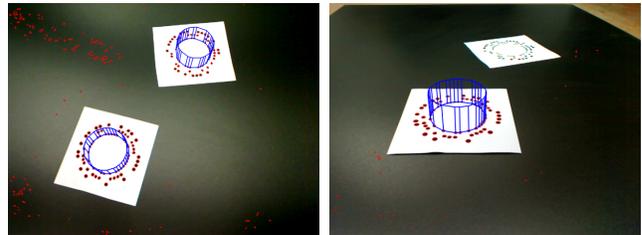

Figure 10. Recognition fails when the marker is angled and far away from the camera and the ellipses blends together.

4. Conclusions

The proposed fiducial marker system exhibits several advantages over the current range of designs. It is both very resistant to occlusion thanks to its code-theoretic design, and offers very high accuracy in pose estimation. In fact, our experimental validation shows that the precision of the pose recovery can be about an order of magnitude higher than the current state-of-the-art. This advantage is maintained also with a significant level of artificial noise, blur, illumination gradient and with up to 70% of the features occluded. Further, the design of the marker itself is quite flexible as can be adapted to accommodate a larger number of different codes or an higher resilience to occlusion. In addition, the identity between the features to be detected and the pattern to be recognized leaves plenty of space in the marker interior for any additional payload or even for placing a physical object for reconstruction tasks. Finally, while slower than other techniques, this novel method is fast enough to be used in real-time applications. Of course those enhancements do not come without some drawbacks. Specifically, the severe packing of circular points can lead the ellipse detector to wrongly merge features at low resolution. This effectively reduces the maximum distance at which a target can be recognized. However, 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.

References

[1] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 1st edition, 2008. 114

[2] Q. Chen, H. Wu, and T. Wada. Camera calibration with two arbitrary coplanar circles. In *European Conference on Computer Vision - ECCV*, 2004. 115

[3] 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. 113

[4] 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. 113

[5] 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 (CVPR'05)*, CVPR '05, Washington, DC, USA, 2005. IEEE Computer Society. 114

[6] M. Fiala. Designing highly reliable fiducial markers. *IEEE Trans. Pattern Anal. Mach. Intel.*, 32(7), 2010. 114

[7] 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, 1991. SPIE. 113

[8] D. Q. Huynh. The cross ratio: A revisit to its probability density function. In *Proceedings of the British Machine Vision Conference BMVC 2000*, 2000. 114

[9] J. Kannala, M. Salo, and J. Heikkilä. Algorithms for computing a planar homography from conics in correspondence. In *British Machine Vision Conference*, 2006. 118

[10] 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, 1999. IEEE Computer Society. 114

[11] V. A. Knyaz, H. O. Group, and R. V. Sibiryakov. The development of new coded targets for automated point identification and non-contact surface measurements. In *3D Surface Measurements, International Archives of Photogrammetry and Remote Sensing*, 1998. 113

[12] 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. 114

[13] J. H. V. Lint. *Introduction to Coding Theory*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998. 116, 117

[14] 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.*, 21, 2010. 114

[15] 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. 114

[16] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, Washington, DC, USA, 2002. IEEE Computer Society. 113

[17] J. Ouellet and P. Hebert. Precise ellipse estimation without contour point extraction. *Mach. Vision Appl.*, 21, 2009. 118

[18] 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, 2000. ACM. 114

[19] L. Teixeira, M. Loaiza, A. Raposo, and M. Gattass. Augmented reality using projective invariant patterns. In *Advances in Visual Computing*, volume 5358 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008. 114

[20] 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, 1998. IEEE Computer Society. 114

[21] A. van Rhijn and J. D. Mulder. Optical tracking using line pencil fiducials. In *Proceedings of the eurographics symposium on virtual environments*, 2004. 114

[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*, 99, 2010. 114, 119

[23] 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, 2004. ACM. 115