Efficient Line-Based Visual Marker System Design with Occlusion Resilience

Abdallah Bengueddoudj[1*], Foudil Belhadj[1], Yongtao Hu[2], Brahim Zitouni[3], Yacine Idir[1], Ibtissem Adoui[1], Messaoud Mostefai[1]
[1]LMSE Laboratory, Mohamed El-Bachir El-Ibrahimi University of Bordj Bou Arreridj, Algeria
[2] Guangdong Virtual Reality Technology Co., Ltd., Guangzhou, China
[3]Private Optical Institute of Bordj Bou Arreridj, Algeria
E-mail : abdallah.bengueddoudj@univ-bba.dz
*Corresponding author

Keywords: visual markers, homography, perspective distortion, occlusion, pose estimation

Received: October 2, 2024

Today, the most widely used visual markers, such as ArUco and AprilTag, rely on square pixel arrays. While these markers can deliver satisfactory detection and identification outcomes, they remain vulnerable to corner occlusion despite incorporating corrective codes. Conversely, line-based markers offer increased resilience against occlusions but are typically constrained in terms of codification capacities. The markers developed in this research leverage linear information to propose a pyramidal line-based structure that exhibits robustness to corner occlusion while providing enhanced coding capacities. Moreover, the projective invariance of the constituent lines enables the validation of a homography-less identification method that considerably reduces computation resources and processing time. We assembled an extensive test dataset of 169,713 images for evaluation, including rotation, distances, and different levels of occlusion. Experiments on this dataset show that the proposed marker significantly outperforms previous fiducial marker systems across multiple metrics, including execution time and detection performance under occlusion. It effectively identifies markers with up to 50% occlusion and achieves identification at a resolution of 1920×1080 in 17.20 ms. The developed marker generation and identification, as well as an extensive marker Database, are publicly available for tests at https://github.com/OILUproject/OILUtag

Povzetek: Predstavljen je izboljšan sistem vizualnih oznak, ki temelji na linijah in je odporen na delne zakritosti. Predlagana je piramidalna strukturaoznak, ki omogoča večjo kodirno zmogljivost in robustnost proti zakritju vogalov. Razvili so metodo identifikacije brez uporabe homografije, kar zmanjšuje računske zahteve in čas obdelave.

1 Introduction

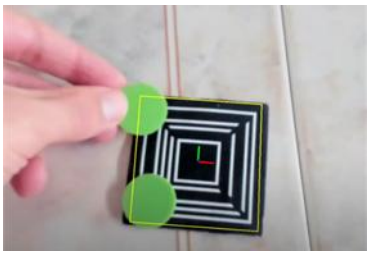
Visual markers are artificial graphical codes representing numerical (or textual message) information that can be associated with objects to be uniquely identified. Computer vision applications use these tags to simplify the automatic perception of objects inside a scene and make their localization more precise. These are widely used in product labeling and tracking, robotics localization and mapping [1], camera calibration and pose estimation [2-3], augmented reality applications [4], automatic navigation [5] and medical positioning [6].

Today, the most prevalent visual markers, such as April [7] and ArUco [8] Tags, utilize square pixel arrays. Although these markers often yield satisfactory detection and identification results, they remain susceptible to external corners occlusion despite the inclusion of error correction codes. In contrast, line-based markers offer greater resilience against occlusions but are often constrained in coding capacities [9]. Recently, Chahir et al. [10] introduced a novel line-based marker called the OILU marker, addressing codification limitations. This marker utilizes groups of pyramidal-shaped lines to create highly distinguishable 2D markers (Figure 1.a).

While offering significant advantages in coding capabilities, the developed identification method, which relies on a time-consuming level set technique [11], slows down processing, particularly in scenarios where multiple markers are in the camera's field of view. The reported average processing time is approximately 40 ms per marker, making this solution unsuitable for constraining real-time applications. In addition, the proposed scheme (marker design and identification method), does not solve the problem of external corners occlusion, for which most square markers remain ineffective. In fact, if just one corner of these markers is occulted, the detection fails.

To overcome these challenges, we propose in this paper, a less computational identification method, based on cumulative histogram analysis that allows reducing processing times by almost half compared with the work of Chahir et al. [10]. However, as the method relies only on the external marker’s corners for localization, it also remains vulnerable to external corners occlusion. Moreover, as the identification scheme integrates

(a) Classical OILU Marker embedding the decimal number 6789



(b) Improved OILU Marker

Figure 1: Classical OILU marker embedding the decimal number 6789 and improved OILU Marker identification and pose estimation under corners occlusion.

homography transform in its processing, computation performances decrease as the number of markers within the camera's field of view increases.

To remedy these limitations, a revised marker design (Figure 1.b), as well as a dedicated homography-less identification approach, are proposed. The adopted identification scheme exploits the marker's local properties to switch from a line-based representation to a more accurate and relatively fast dot-based one.

Deep tests on real and synthetic images highlight the performance and robustness of the proposed solution against challenging conditions, with a particular focus on corners occlusion. Despite this, the homography transform remains applicable for pose estimation, where improved markers demonstrate superior performances compared to state-of-the-art markers

In summary, the main contributions of this paper are:

- The layout design of the OILU Tag has been enhanced to offer more robustness to occlusion and overlapping objects.
- The proposal introduces a low computational homography-less identification method. The average execution time has been considerably reduced for both desktop and mobile architectures, making it suitable for constraining real-time applications.
- A dedicated OILU Tag Generator as well as a huge database are made available for comparative tests with the well-known state-of-the-art visual Tags.

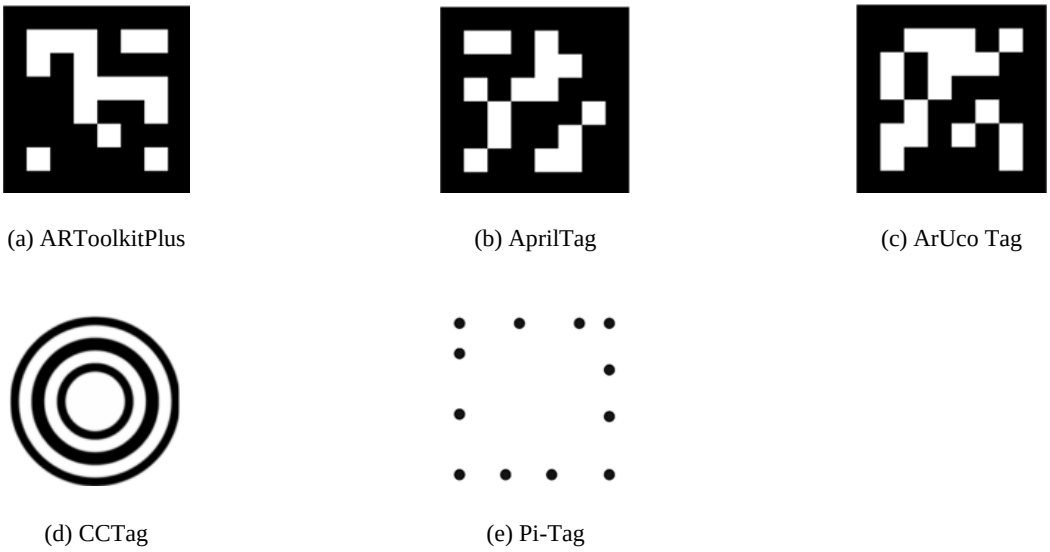
The remainder of this paper is organized as follows: Section 2 provides a quick literature review on well-known fiducial markers. Section 3 briefly presents the OILU code basics and highlights its key strengths as being an efficient visual marker. Sections 4 describe our primary OILU marker identification scheme, followed in Section 5 by the presentation of a revised marker design, as well as its validated homography-less identification approach. In section 6, extensive tests are conducted on real and synthetic images. Finally, section 7 concludes the paper with interesting perspective views.

2 Related works

There are many conceptions of visual markers in the literature (Figure 2). These can be clustered into three main categories: square-based, line-based, and dot-based tags. The first category regroups all QR-like tags that encode binary information in black/white cells arranged in square grid layouts. ARToolKit [12] is the oldest fiducial marker proposed for AR applications. It consists of a black-bordered square inside, which is embedded in a known image as a payload. Its limitation resides in the matching method that uses image correlation techniques to detect the embedded pattern. ARToolKitPlus and ARTag [13-14] are improved versions released to overcome these limitations. They use binary-coded patterns to encode the embedded identifier. Furthermore, the ARTag introduces additional information as an error-correction payload. Based on ARTag's idea, many efficient square markers were proposed, among them April Tag [7] and ArUco Tag [8], which became ubiquitous in the AR field. Both allow generating of user-customized dictionaries using some heuristics to maximize some criterion such as inter-marker distance and the number of bit-transitions.

Recently, a new square-like TopoTag was introduced by Yu et al. [15]. It offers a highly customizable marker shape. The fundamental structure of the marker consists of a black frame with black squares positioned on a white background. One notable advantage of TopoTag is its variable dictionary size. The authors claim that generating the dictionary is significantly faster compared to similar marker systems like April and ArUco Tags. Based on AprilTag, [16] proposed ChromaTag by using different colors to represent the internal bits to make the detection easier and speed up its decoding. Line-based markers apply some measurements on the basic forms like line-thicknesses and angle sizes to encode the elementary information. Usually, markers in this approach are robust against bad acquiring conditions such as blurring and variation in lighting. They perform well in occlusion situations. Based on the classical linear bar code, Calvet et al. [17] proposed a circular version called CCTag, in which the lines have been substituted by circles with different thicknesses.

Dot-based tags [18-19] enable the developing of projective invariants fiducial marker systems based on cross ratios computation. Even though these markers exhibit higher accuracy in camera calibration and pose



(a) ARToolkitPlus (b) AprilTag (c) ArUco Tag

(d) CCTag (e) Pi-Tag

Figure 2: Examples of well-known visual markers.

Table1: Main markers specifications

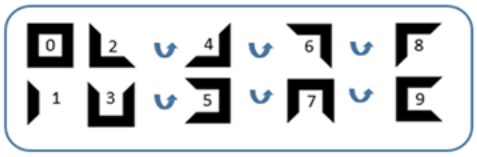
Marker	Family	Shape	Scalability	Key Findings	Flexibility
AprilTag [7]	Qr-like	Square	Limited	High dictionary size (5329); widely used in robotics and AR systems	Low occlusion resilience; limited scalability; no flexibility
ArUco [8]	Qr-like	Square	Limited	Moderate dictionary size (250); simple implementation	Low occlusion resilience; limited scalability; no flexibility
ARToolKit [12]	Qr-like	square	Limited	Developed the first generation of AR marker tracking tools	lacks flexibility and high computational complexity
CCTag [17]	Bar-like	Circular	Limited	Easy detection of circular shapes	Very limited dictionary size (39); low occlusion resilience; no flexibility
Pi-tag [18]	Dot-like	Square	Limited	Balanced dictionary size (300); efficient marker detection	Low occlusion resilience; limited scalability; no flexibility

estimation, they offer a limited number of distinctively recognizable patterns [20].

OILU Tag [10] is a distinct type of fiducial marker based on the two initial categories. It distinguishes itself from other fiducial markers in two main aspects: firstly, humans and machines can read it. Secondly, it exclusively employs lines as primary patterns to encode the elementary information. Moreover, the OILU marker introduces a novel approach to layout design, ensuring user accessibility and ease of implementation. It can even be accurately reproduced by hand drawing, eliminating OILU Tag [10] is a distinct type of fiducial marker, based on the need for specialized tools or intricate encoding schemes, making it ideal for a wide range of applications. Table 1 presents well-known markers along with their features, such as shape and dictionary size.

In summary, existing visual markers have significant limitations. Square-based markers, such as AprilTag and ArUco, are particularly susceptible to corner occlusion, resulting in poor performance even with minimal obstruction. While line-based markers are more resilient to occlusion, they are hindered by limited coding capacity and high computational requirements, especially when dealing with numerous markers in the field of view. Moreover, these markers often rely on complex homography transformations, reducing their efficiency.

The new marker design overcomes these challenges with several key improvements. By utilizing lines as the primary pattern, OILU markers offer inherent redundancy, making them more resistant to occlusion and blurring compared to dot- or square-based markers. Additionally,



(a) Decimal OILU Symbols representation

(b) Pyramidal OILU representation of the decimal number 1962

Decimal	OILU Symbols	Symbol's components			
		Seg1	Seg2	Seg3	Seg4
0		1	1	1	1
1		1	0	0	0
2		1	1	0	0
3		1	1	1	0
4		0	1	1	0
5		0	1	1	1
6		0	0	1	1
7		1	0	1	1
8		1	0	0	1
9		1	1	0	1

(c) OILU Symbols binary codification based on the presence (1)/absence (0) of the composing segments {Seg1, Seg2, Seg3, Seg4}

Figure 3: OILU Symbols representation. The whole symbols are incorporated in a square to delimit their area in a real-world scene

lines serve as effective separators between contrasting regions, improving marker detection and recognition.
OILU's unique pyramidal line-based structure enhances occlusion resilience, ensuring reliable detection even in difficult conditions. The homography-less identification method reduces computational demands, making it more suitable for real-time applications. The shift from line-based to dot-based representations also boosts detection accuracy and processing speed. These innovations position the OILU marker as a superior alternative to existing visual markers, striking a balance between robustness, efficiency, and practicality.

3 OILU markers basics

OILU markers, as described in [10], are based on a set of four basic symbols {O, I, L, and U}, corresponding respectively to digits zero, one, two, and three (Figure 3 (a). The remaining decimal symbols, related to digits {4, 5, 6, 7, 8, 9}, are obtained by successive counter-clockwise rotations of the two symbols L and U. The important feature of these symbols is their ability to be concatenated in a pyramidal fashion, producing multi-faceted numbers that can be exploited as visual markers (Figure 3 (b)). Each OILU symbol is coded in bi-nary according to Figure 3 (c). In the following, we will detail our improved identification approach based on cumulative histograms analysis. Compared to the level set method presented in [10], the adopted approach is relatively simple and computationally efficient. It operates on classical OILU markers and incorporates homography in its processing [21, 22].

4 Standard OILU markers detection and identification

To facilitate detection, the visual OILU markers are designed with black-outlined segments on a white background (or inversely). The identification process follows the standard computer vision processes, which involves three key stages: pre-processing, code detection, and decoding. The complete process presented in Figure 4 is as follows:

4.1 Preprocessing

The primary objective of the pre-processing stage is to enhance the quality of the captured images, ensuring they are optimized for subsequent stages. To achieve this, classical image processing filters can be applied [23], though modern cameras often capture high-resolution images. In real-time applications, there is a necessary trade-off between speed and accuracy. Downsampling the captured images enables quick noise filtering and reduces the execution time, especially in the subsequent stages.
The output of this stage is an improved grayscale image (as shown in Figure 4 (b)). Its goal is to localize all possible quadrilaterals eligible to be square-OILU markers in the grayscale image. The process comprises three main steps:

4.2 Eligible markers detection

4.2.1 Image Thresholding

The first step after obtaining the enhanced grayscale image is to binarize it, separating the objects in the image from the background. This makes the extraction of contours possible in the subsequent step. Several methods can be used for binarization [23]. The simplest method is direct thresholding, where a global threshold is applied; however, this method performs poorly on images with

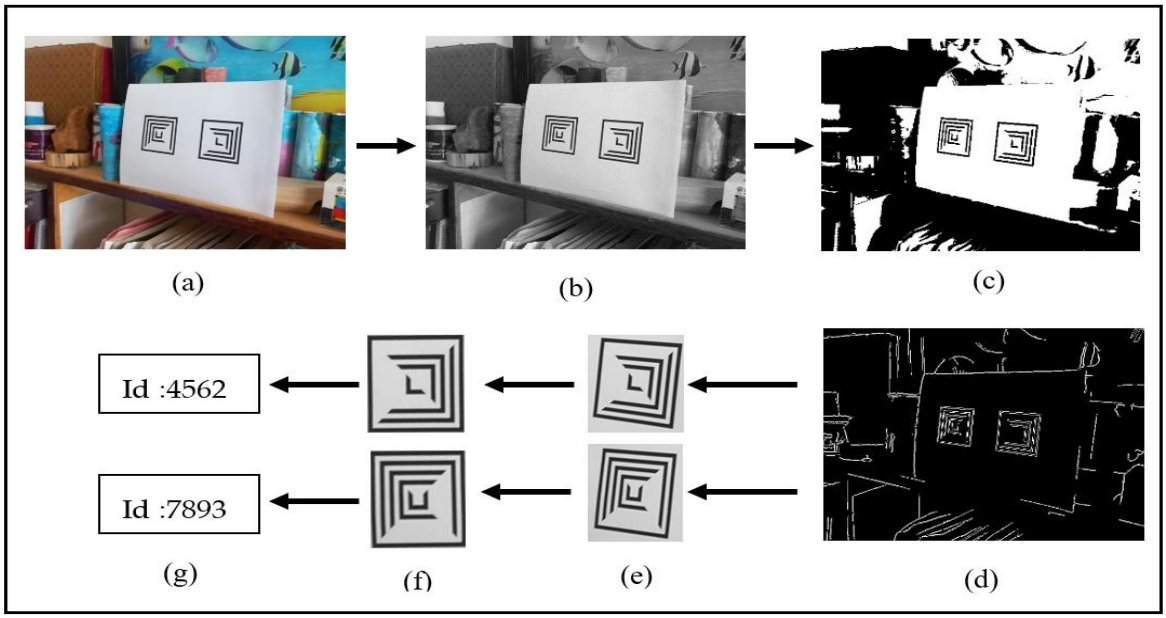


Figure 4: OILU Marker Detection process. (a) Input image acquiring. (b) input image processed and grayscale converted. (c) Binarized image. (d) Contours extracted. (e) Eligible markers extracted. (f) Perspective correction using homography. (g) Markers decoded.

multimodal histograms. The Canny method can be used, but it is time-consuming for real-time applications. For better performance, we utilize a local adaptive thresholding method, which is robust to varying lighting conditions and does not depend on a global threshold choice. Figure 4 (c) depicts the resulting binarized image.

4.2.2 Contour extraction

Given the square shape of the OILU Tag, we search for all potential quadrilateral shapes in the binarized image that could correspond to an OILU marker. To accomplish this, we first extract the contours of the image by tracing the transitions between black and white pixels, as described in [24]. Next, we approximate the obtained contours to the nearest polygonal shape using the Douglas-Peucker algorithm [25] (as shown in Figure 4 (d)). Only convex shapes with four corners are retained (Figure 4(e)). Some refinement steps are necessary to eliminate contours that are too small, too large, or too close to Each other [26].

4.2.3 Candidate markers determination and perspective adjustment

Although we retain all convex quadrilaterals with four corners in the previous step, not all of them are regular squares. Some may be subject to 2D transformation constraints such as rotations or perspective distortions. To correct these irregularities, a homography is applied to the sub-image framed by the quadrilateral. Once corrected, each obtained sub-image is resampled to a canonical grayscale image of size $Wc \times Wc$ using linear interpolation. The output of this step is a list of candidate square-shaped marker images (as depicted in Figure 4 (f)).

4.3 Marker Identification

Each candidate marker in the obtained list needs to be processed to confirm its content as an OILU marker and read its embedded identifier. As previously mentioned, each identifier digit is encoded in a separate layer using four segments that reflect the OILU symbolic. The challenging part is to identify the position of each layer, locate each segment within it, and extract its binary content, particularly in critical situations such as occlusions and noise. More formally, let K be an integer with N decimal digits, and M its corresponding OILU code. The segment-based binary codification of M is:

$$M = \left\{\left(s_0^i, s_1^i, s_2^i, s_3^i, s_4^i\right)\right\}_{i=1}^{N}, s_j^i \in \{0,1\}\, for\ j = 1..4 \quad (1)$$

The size of the embedded identifier (N), which corresponds to the number of layers, is unknown beforehand. Furthermore, no assumptions are made regarding the thickness of the segments, whether they are equal or not. When the segments are of equal width, the binary square image can be divided into a matrix of the same width and height as the segment width to isolate the segments easily. However, designing an OILU marker with different segment thicknesses and inter-layer space widths makes it more flexible and robust to a wide range of distortions, occlusions, and noise. In the subsequent paragraphs, we will consider this last case, which is more challenging. The decoding procedure, illustrated in Figure 5, involves several steps:

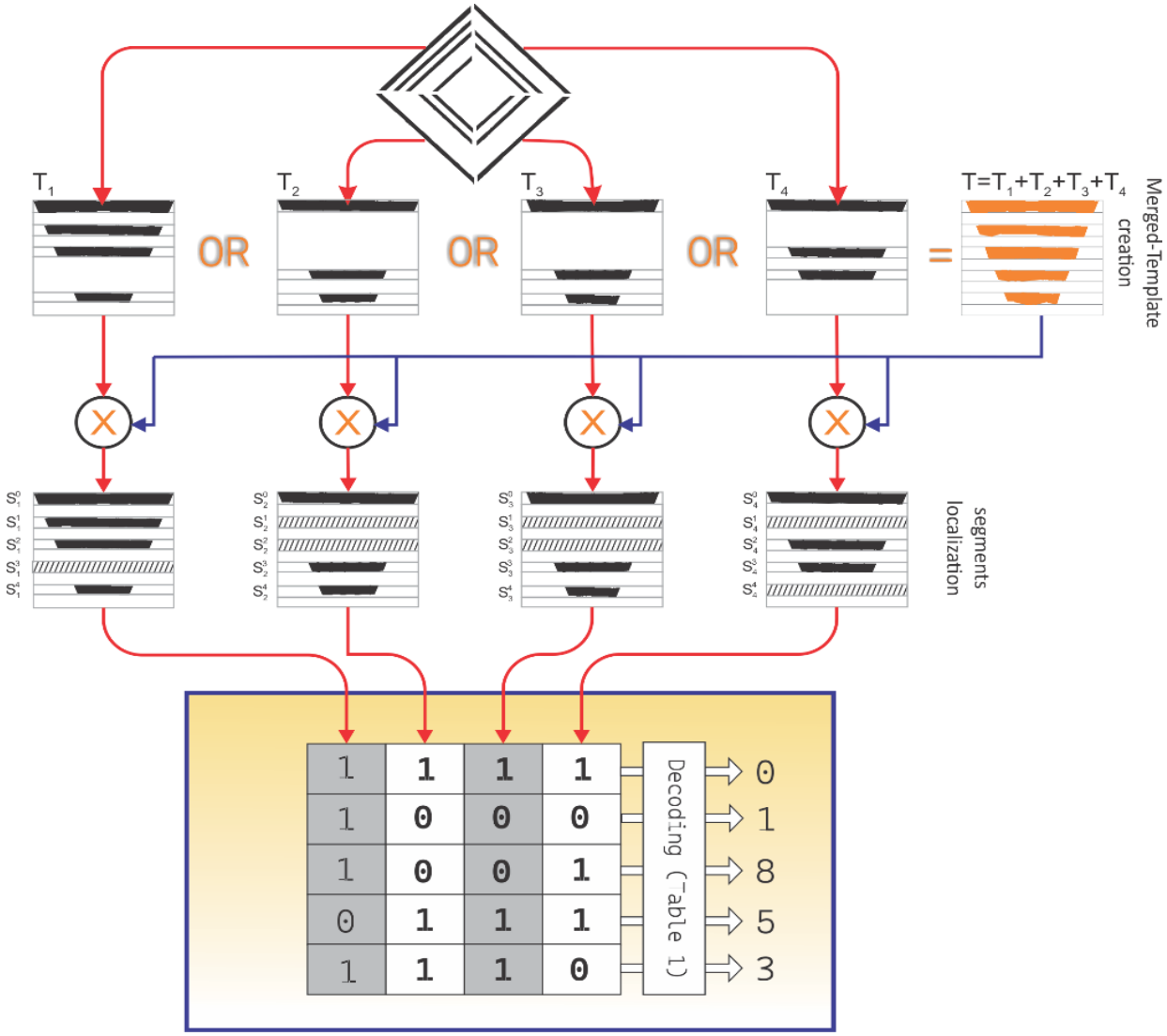


Figure 5: Decoding process: Each layer in a given OILU marker contains at least one black segment. Performing a bitwise-or operator on the four triangles (sectors) constituting the OILU -marker results in a merged template that contains a black segment on each layer. The merged template allows delimiting each black/white segment in each individual triangle.

4.3.1 Binarization

OILU tags exhibit bimodal histogram characteristics, making Otsu's thresholding method [27] the ideal choice for generating binarized images. This method determines the optimal threshold value to distinguish between the predominantly white background and the typically black OILU segments. In the resulting binarized image (Figure 4(c)), pixels within the segments (i.e., the region of interest or ROI) are assigned a value of '1' while all other pixels are assigned a value of '0'.

4.3.2 Layers extraction and black segments localization

The binary image is first divided into four triangles or sectors, denoted as T1, T2, T3, and T4. Each triangle Ti comprises a set of alternating black and white bands that contain the encoded segments. A black band indicates a binary one '1', while a white band could represent zero '0' or multiple consecutive zeros '0' (as illustrated in Figure 5). To locate the segments within the image, we utilize a useful property of the OILU marker that states 'each layer contains at least one black segment'. Therefore, combining all the triangles by performing a bitwise-OR operation between their contents yields a template triangle T (2) containing the exact number of black segments equal to the number of digits N in the encoded identifier (as depicted in Figure 5).

$$T = T1 + T2 + T3 + T4 \quad (2)$$

The merged triangle T plays the role of a template guide that allows delimiting all the black/white segments in each layer by analyzing its horizontal and vertical cumulative

Table 2: OILU decoding matrix. Each column corresponds to one triangle in which each segment is coded in binary (s_i^j=1 means the segment is detected as being black)

T1	T2	T3	T4	Lookup Figure 3 (c)
s_0^1	s_0^2	s_0^3	s_0^4	$digit_0$ (must equal 1111)
s_1^1	s_1^2	s_1^3	s_1^4	$digit_1$
⋮	⋮	⋮	⋮	⋮
s_{N-1}^1	s_{N-1}^2	s_{N-1}^3	s_{N-1}^4	$digit_{N-1}$
s_N^1	s_N^2	s_N^3	s_N^4	$digit_N$

histograms (respectively HCH and VCH). The horizontal histogram HCH is the sum projection of pixel values along all rows inside the triangle T.

$$HCH_j = \sum_{k=1}^{w_c} \mathrm{T}(j,k)\,; j = 1..W_c/2 \quad (3)$$

It allows localizing the black segments by following the black-white transitions. Indeed, black segments coincide with high ridges (peaks) in the HCH, while white ones constitute low valleys. To handle occlusion situations and to be robust against noise, a percentage threshold $\omega_1 = 2/3$ regarding the whole line is set up to decide whether a horizontal-histogram value is black or white.

$$HCH_j \geq (\mathrm{w}_c - 2 * j) * \omega_1 \Rightarrow T(j) \equiv black\ line \quad (4)$$

It is worth noting to mention that ω_1 is dependent on the row position; outer rows correspond to high values of ω_1 and vice-versa. The VCH is the vertical projection of T over all columns; it allows the detection of the number of black bands, confirming the horizontal histogram analysis results.

$$VCH_j = \sum_{k=1}^{w_c/2} \mathrm{T}(k,\mathrm{j})\,; j = 1..w_c \quad (5)$$

The clustering of many adjacent black (respectively white) rows in the HCH constitutes a black (respectively white) segment, provided that the number of rows exceeds a threshold $\omega_2 = 25\%$. After creating the merged template and locating its segments, we utilize it to determine the position of each segment within the four triangles.

4.3.3 Marker validation

To confirm that the embedded data is an OILU code, we only need to verify that the marker satisfies the following criteria, which serve as an OILU signature:

- The strict alternation of bands: the merged triangle T comprises alternating black-white segments. The most outer (starting) band is black, and the most inner (ending) is white. The number of all bands is always even.
- Each black band in a triangle must correspond to a black band in the merged triangle.
- Each segment must be connected (no small fragments).

4.3.4 Marker decoding

To decode the content of the validated marker, we follow the reverse process of the encoding procedure (as shown in Figure 5, decoding step), which involves the following steps:

We affect the value "1" for each black segment and "0" for each white one, starting from the most outer segment to the most inner. Each triangle (T_i) i=1..4 is composed of N+1 segments:

$$\mathrm{T_i} = \left\{s_i^k\right\}_{k=0}^{N}, s_i^k \in \{0,1\}\ for\ i = 1..4 \quad (6)$$

Next, we concatenate the binary values inside each triangle Ti to form a binary string:

$$str_i = s_0^i s_1^i .. s_{N-1}^i s_N^i, for\ i = 1..4 \quad (7)$$

After that, we arrange the four binary strings vertically to form a decoding matrix (Table 2) starting with the left triangle and going counterclockwise (left, bottom, right, then upper). Each line of the decoding matrix represents a digit in the identifier whose decimal value can be obtained from the OILU codification table (Figure 3-c). These aforementioned steps are repeated for all eligible markers, and only the validated markers that have their IDs and Cartesian coordinates within the original image are retained after the detection process.

4.4 Processing time required for standard OILU markers identification

The identification scheme described in section 4 has been implemented and tested on a typical Laptop equipped with a 2.4 GHz Intel Core i7 processor with 16 GB RAM, running Windows 10. The processing time can be divided into three main steps: (1) finding marker candidates (including image processing, contours extraction and eligible squares determination), (2) perspective correction of all candidates, and (3) markers validation. The execution of the first step can be affected by the size S of

Table 3: Average processing time.

Architecture	Step	Proposed method		(Chahir et al., 2021) method
		Average time per step Image (640x480)	Total time /candidate	Total time /candidate
Typical laptop	1	17.33 ms	19 ms	40 ms
	2	1.24 ms		
	3	0.43 ms		
Typical Android smartphone	1	22.08 ms	25.06 ms	Not reported
	2	1.87 ms		
	3	1.10 ms		
Step 1 : Finding Marker Candidate – Step 2 : Perspective corrections – Step 3 : Marker validation				

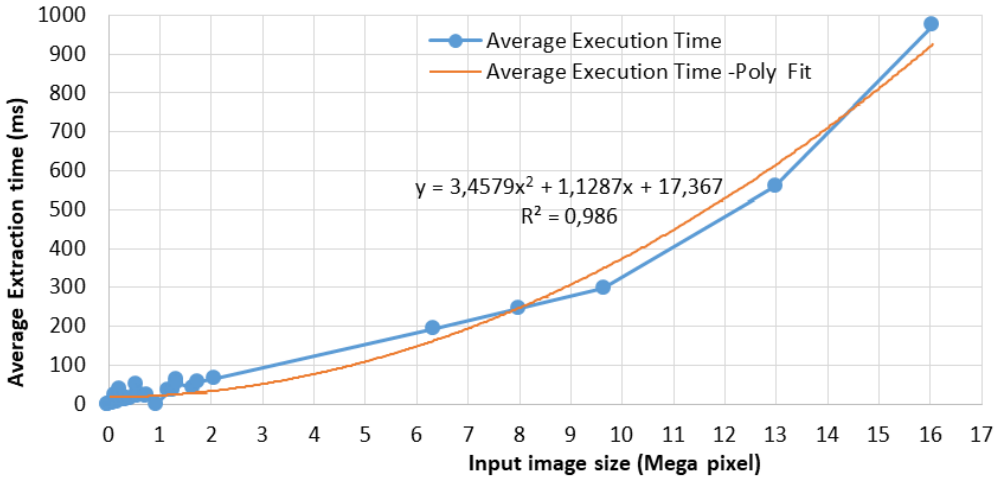


Figure 6: Evolution of the average detection time of the OILU code in function of the input-image size with its polynomial distribution fit.

the input image and the complexity of its texture in terms of contained contours, while the processing of the second and the third steps only depends on the marker canonical size $w_c \times w_c$. Table 3 resumes the average execution time of each step taken for multiple input images of size $S = 640 \times 480$ and canonical size of 256×256 pixels.

Figure 6 shows the evolution of the average extraction time in function of the image size and its best distribution fit which seems to be a polynomial distribution (with R2 goodness-of-fit = 0.986). As observed, the detection time increases significantly, reaching up to one second, when the input image has a high resolution.

While the adopted scheme reduces processing times by nearly half compared to the approach by Chahir et al. [10], it still depends on the external corners of the markers for localization, rendering it susceptible to occlusion. Additionally, the integration of homography transformations in the identification process leads to reduced computational performance as image resolution increases or as more markers appear within the camera's field of view.

In the following, an improved OILU marker system design is proposed (Figure 7). It involves enclosing the embedded identifier within two nested square-like quadrilaterals, enabling efficient marker detection even when the external marker's corners are occluded. The developed identification method considers OILU numbers as groups of locally parallel segments, treating them separately without the need for a homography transform, thereby reducing computation resources and minimizing processing time.

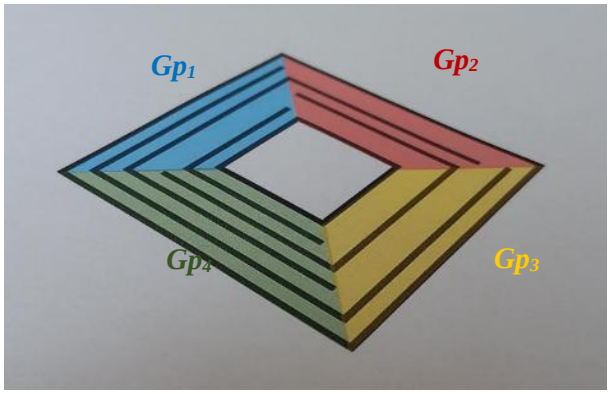


Figure 7: Groups of locally parallel segments.

5 Improved OILU markers system design

Common, well-known problems with state-of-the-art markers include detection failures when their corners are occluded, and lack of size adaptation to the camera's field of view (FOV), especially when the camera is in motion. This is evident, for example, when an autonomous drone attempts a landing based on its on-board camera. These issues have been addressed in various works [2-28-29]. An interesting approach presented in [2] involves designing fractal markers composed of imbricated quadrilaterals. In addition, being multi-scale markers, the latter are robust to partial occlusions. This inherent structure is characteristic of OILU markers, which are made up of nested square symbols, allowing their structure to be customized to overcome the above-mentioned problems. Hence, the adopted structure (Figure 8) is as follows:

- Two imbricated inner/outer square like – quadrilaterals as marker delimiters.
- A group of disconnected segments to embed the marker identifier

(a) $Id_1 = 0000$ (b) $Id_2 = 6819$ (c) $Id_3 = 2372$

Figure 8: Example of OILU markers with fixed inner and outer squares embedding different identifiers. Embedded symbols are drawn with disconnected segments (without corners) to distinguish them from the inner/outer quadrilaterals.

This arrangement simplifies marker detection, even in complex backgrounds, by focusing on identifying nested, similar square-like quadrilaterals within a filmed scene. This specific pattern streamlines OILU marker detection while filtering out non-OILU quadrilaterals. Furthermore, the nested structure enhances the markers' resilience to partial occlusion, which typically affects the outer quadrilateral. Since these imbricated quadrilaterals share a similar structure, any partial occlusion of the outer quadrilateral can be approximated and reconstructed through uniform rescaling. This advantageous feature is extensively leveraged in the experimental section to improve marker detection and pose estimation, even under occlusion.

Figure 9: OILU markers detection within complex backgrounds.

(a) Computed railway interlines ratios (b) Real OILU markers identification based on cross ratios computation

Figure 10: Analysis of Railway Interline Ratios and OILU Marker Identification

5.1 Candidate marker's location

As mentioned, OILU markers are composed of two imbricated square-like quadrilaterals.
Such composition eases markers' location tasks even within a complex background (Figure 9). For more selectivity, the fixed inner/outer quadrilateral's surface ratio is used to eliminate surrounding non-OILU quadrilaterals.

5.2 Marker identification

A deep evaluation of the perspective distortion levels within the adopted markers shows that these are more significant between the different groups of parallel segments (Gp1, Gp2, Gp3 and Gp4) than within the same group (Figure 7). Indeed, parallel lines in the same group remain locally parallel, even if the marker is acquired in perspective. In another way, each group of lines can be considered as railway ties (Figure 10 (a)) for which computed Euclidean length ratios remain invariant to perspective changes [30]. The challenge here is exploiting such ratios to generate the related embedded sub-codes without homography.
The adopted approach involves crossing the composing groups of lines (Gpi) with a virtual line centred on the middle of the marker (Figure 10 (b)). Crossed lines sections are then used to locate the corresponding cross points and generate associated sub-codes. The quantity of retained cross-point sets within each group can range from one to multiple, depending on the desired level of resilience against distortions, particularly occlusion. For example, a single set per group is adequate for marker identification when markers are fully visible. However, in scenarios where parts of the marker are obscured, multiple sets from various regions are required to confirm the most common ones. Notably, identification may fail even with multiple selected regions in case of significant occlusion. Deep tests in the experimental section will show the accuracy and limits of this approach. Globally, the adopted OILU marker code generation process is as follows: First, for each group Gpi (i= 1 to 4), the following metrics are computed:

Group's metrics computation

- *the number (N) of cross points,*
- *the (N -1) inter-dots Euclidian distances {d_j, j=1 to N -1},*
- *the group band width* $W_i = \sum_{j=1}^{N-1} d_j$,
- *the average dots spacing* $A_i = W_i/S$, *with S the number of code symbols,*

the ratios $R_j = round\ (d_j/A_i)$.

Cross points positions are estimated according to their computed ratios Rj and marker's format (number of embedded symbols). In case of a four symbols marker, the possible configurations to be tested are as follows:

Cross points position estimation

R_j = round (d_j/A_i) // with Ai, the average cross points spacing of Gp_i
if *R_j = 1 the corresponding points are adjacent // case (a)*
else if *R_j = 2 related points are separated by one empty space // case (b)*
else if *R_j = 3 related points are separated by two empty spaces // case (c)*
else *related points are separated by three empty spaces // case (d)*
end

The basic example (Figure 11) illustrates where a group's number of cross points equals three (two inner/outer boundary points and one symbol cross point). The number of inter-cross points Euclidean distances is equal to two (d_1, d_2). Computed metrics are: $W_i = d_1 + d_2$; $Ai = W_i/4$; $R_1 = round\ (d_1/A_i)$. Therefore, the presented red symbol cross point will be in one of the four cases $\{(a), (b), (c)\ or\ (d)\}$, according to R_1 value, equal to 1, 2, 3 or 4.
A second illustrative example (Figure 12) shows three views of a real marker (embedding decimal number 0389). Developed identification method, calculates corresponding cross points coordinates and metrics for each group of segments. It is worth mentioning that since processing is carried out separately on each group of cross

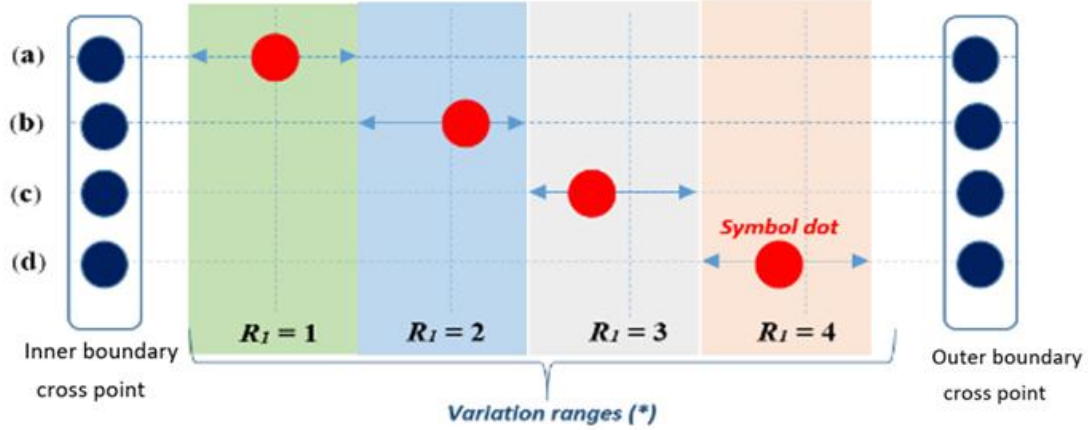
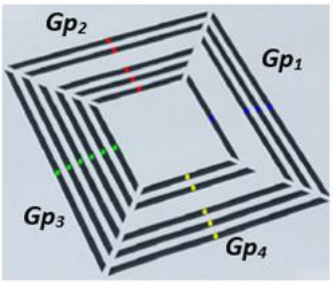
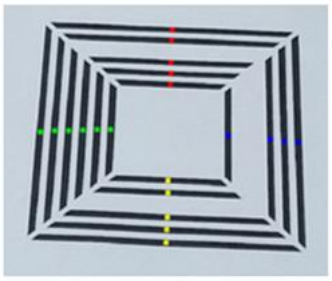


Figure 11: Cross points position estimation, (*) Colored bands delimit the cross-point variation intervals.

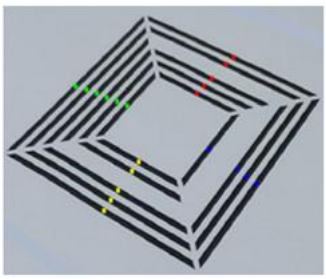
points, perspective distortions have no impact on the computed ratios and, consequently on the related embedded codes, making homography transformation unnecessary for marker identification. In the following, deep tests on real OILU markers are performed to evaluate the correctness and robustness of this approach against leading.



Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	71.67	19.91	22.98	21.32
d_2	23.93	19.93	23.52	43.49
d_3	24.70	39.50	23.19	22.58
d_4	-	20.07	23.65	22.67
d_5	-	-	24.18	-
A_i	24.06	19.88	23.51	22.01
Code	1 1 0 0	1 0 1 1	1 1 1 1	1 1 0 1



Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	68.55	17.93	22.59	19.89
d_2	23.12	17.87	23.14	40.21
d_3	22.99	35.63	22.91	20.38
d_4	-	17.59	23.19	20.78
d_5	-	-	23.17	-
A_i	22.93	17.81	23.00	20.25
Code	1 1 00	1 0 1 1	1 1 1 1	1 1 0 1



Gp_i	Gp_1 (4 dots)	Gp_2 (5 dots)	Gp_3 (6 dots)	Gp_4 (5 dots)
d_1	71.02	20.17	24.19	22.31
d_2	23.84	20.03	24.55	44.60
d_3	23.76	40.18	24.28	22.50
d_4	-	19.61	24.75	23.04
d_5	-	-	24.61	-
A_i	24.27	22.65	24.48	25.30
Code	1 1 0 0	1 0 1 1	1 1 1 1	1 1 0 1

Figure 12: Computed cross points metrics and related embedded codes.

6 Experiments

Tests are carried out on a huge database of synthetic and real markers, with nearly 5000 markers available in three groups of different sizes: small ($5\ cm \times 5\ cm$), medium ($10\ cm \times 10\ cm$), and large ($15\ cm \times 15\ cm$). Printed markers are placed on a rotating support using different types of cameras. Specifically, we employ a high-resolution Logitech camera (Figure 13). Deeper tests with many markers, displayed on a Surface Pro X tablet are also performed to assess the performance of our method under various distortion conditions (Figure 13 (b)). In our tests, we compared the performance of our developed marker with two well-known markers, ArUco and AprilTag. We gathered data for each tag family: 36h12 and 16h3 for ArUco and 25h9 and 36h10 for AprilTag.

Codes for the developed marker (generation/detection) and the OILU database (images and videos) are publicly available for download at the following link: *https://github.com/OILUproject/OILUtag*.

6.1 Marker to camera distance impact

Initially, we evaluate the impact of the marker-to-camera distance on the performances of marker detection. The camera was positioned in front of the marker at different distances d, ranging from 0.2m to 4 m. Obtained identification results are presented in Table 4. Compared with the ArUco and April Tags results, the proposed marker performs less well when using a fixed-focus camera (Logitech in our case). As distance increases, adjacent parallel lines expand, forming a uniform area that prevents accurate identification. This problem can be solved using an autofocus camera, such as a smartphone. Note that after a certain distance (superior than 3.5 m), marker identification became dependent on the camera resolution. The higher the resolution, the better the identification and vice versa.

(a) A high-resolution front camera is used to record video sequences at various distances ranging from 0.2 to 4 meters

(b) A Surface Pro X tablet serves as a display platform to validate the identification scheme across a large database of markers

Figure 13: The experiment setup involves a rotating plate with embedded markers, which is controlled by a stepper motor to accurately capture the markers in a perspective view.

Table 4: Robustness to distance.

Cameras	Distance (m)	ArUco			April			OILU		
		T_1	T_2	T_3	T_1	T_2	T_3	T_1	T_2	T_3
Logitech Camera	2			×			×		×	×
	3					×				
	4	×	×		×			×		
Marker dimension : T_1 (5cm×5cm) – T_2(10cm×10cm) – T_3 (15cm×15cm)										

(a) Failing identification case (β =10)

(b) Successful marker identification (β = 20)

(c) Successful marker identification (β =40)

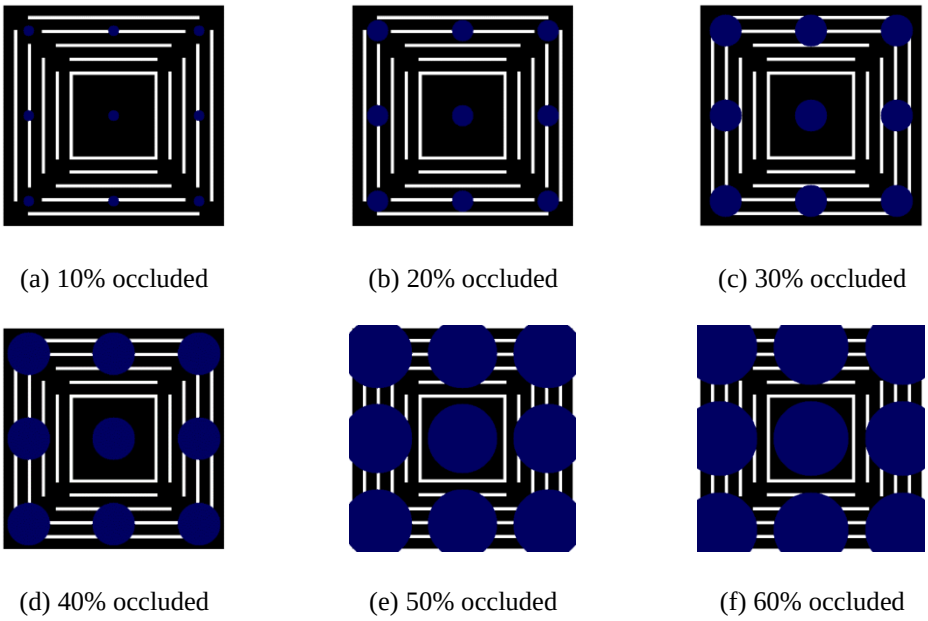
Figure 14: Snapshots of a live video showing different perspective views of a real marker embedding the value 2758

6.2 Robustness to viewing angle

The second test concerned robustness to viewing angle ‘β’. Markers were placed 1 m away from the camera and acquired with varying viewing angles β ∈ [10°: 90°]. The obtained results show that all the codes examined are indeed detected at angles greater than 15° (Figure 14). Beyond this angle, the proximity of neighbouring parallel lines increases, forming a homogeneous region that prevents accurate identification.

6.3 Robustness to occlusion

In these tests, we use a set of 50 unique OILU markers, each marked by a varying number of opaque circles ranging from 1 to 9 (Figure 15). By adjusting the size of these circles across seven different sizes, we generated a total of 3150 test images. The same process is adopted with the well-known April and ArUco Tags. Generated database is evaluated using dedicated exploitation codes. Obtained comparative tests (presented in Table 5) confirm that the suggested marker, characterized by its consistent line-based pyramidal structure, outperforms standard markers in handling difficult occlusion distortions. Identification fails if the occlusion rate exceeds 70% or both inner and outer quadrilaterals are partially occluded. Examples of snapshots from an available live video (Figure 16) show occulted markers identification cases in perspective view.



(a) 10% occluded (b) 20% occluded (c) 30% occluded

(d) 40% occluded (e) 50% occluded (f) 60% occluded

Figure 15: Occlusion tests using a set of 3150 synthetic markers. Opaque variable size circles are used for occlusion.

Table 5: Robustness to occlusion.

Occlusion (%)	Accuracy of the identification (in %)				
	OILU Tag	ArUco (36h12)	ArUco (16h3)	April (25h9)	April (36h10)
10%	100 %	72.67 %	100 %	100 %	100 %
20%	100 %	11.78%	100 %	23.17%	14.22 %
30%	100 %	10.44 %	100 %	17.87	10.89 %
40%	100 %	8.44 %	25.33 %	09.33 %	08.88 %
50%	100 %	1.56 %	17.78 %	04.53 %	01.78 %
60%	100 %	1.56 %	13.83 %	00.00 %	00.00 %

The tests shown in Figure 17 demonstrate the advantages of the OILU marker structure compared to the commonly used ArUco and AprilTag markers for pose estimation. While this task is relatively simple with is no occlusion (as shown in Figure 17 (a)), it becomes much more challenging when part of the marker is blocked. Markers like ArUco and AprilTag often struggle to maintain accuracy in these situations.

OILU markers, however, show strong resilience to partial occlusion, as illustrated in Figures 17 (b), (c), and (d). The design of OILU markers is particularly effective because, even if the outer quadrilateral is partially blocked, the pose can still be estimated using the inner quadrilateral. This feature makes OILU markers well-suited for environments where markers may not always be fully visible. However, OILU markers also exhibit certain limitations. When both the inner and outer quadrilaterals are occluded simultaneously, referred to as severe occlusion, pose estimation is no longer possible. Despite this, OILU markers significantly improve occlusion handling compared to ArUco and AprilTag markers, making them a more reliable choice in many real-world applications.

(a) 15% corner occlusion

(b) 30% corners occlusion

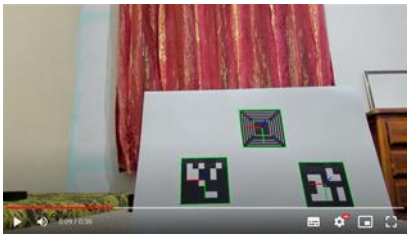
(c) 40% corners occlusion

(d) 50% corners occlusion

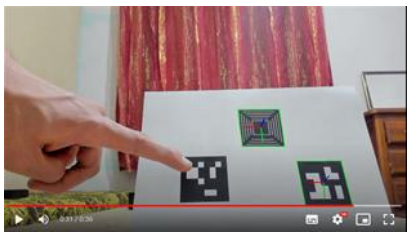
(e) 40% middle segments occlusion

(f) 50% middle segments occlusion

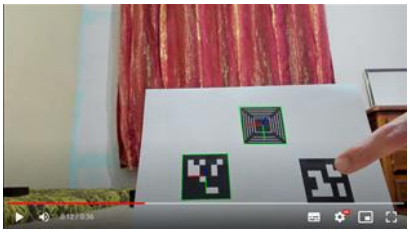
Figure 16: Snapshots of live demo showing occlusion tests with real markers acquired in perspective view.



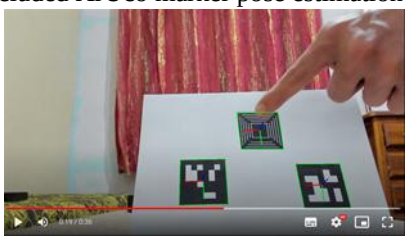
(a) Occluded ArUco marker pose estimation failure



(b) Occluded ArUco marker pose estimation failure



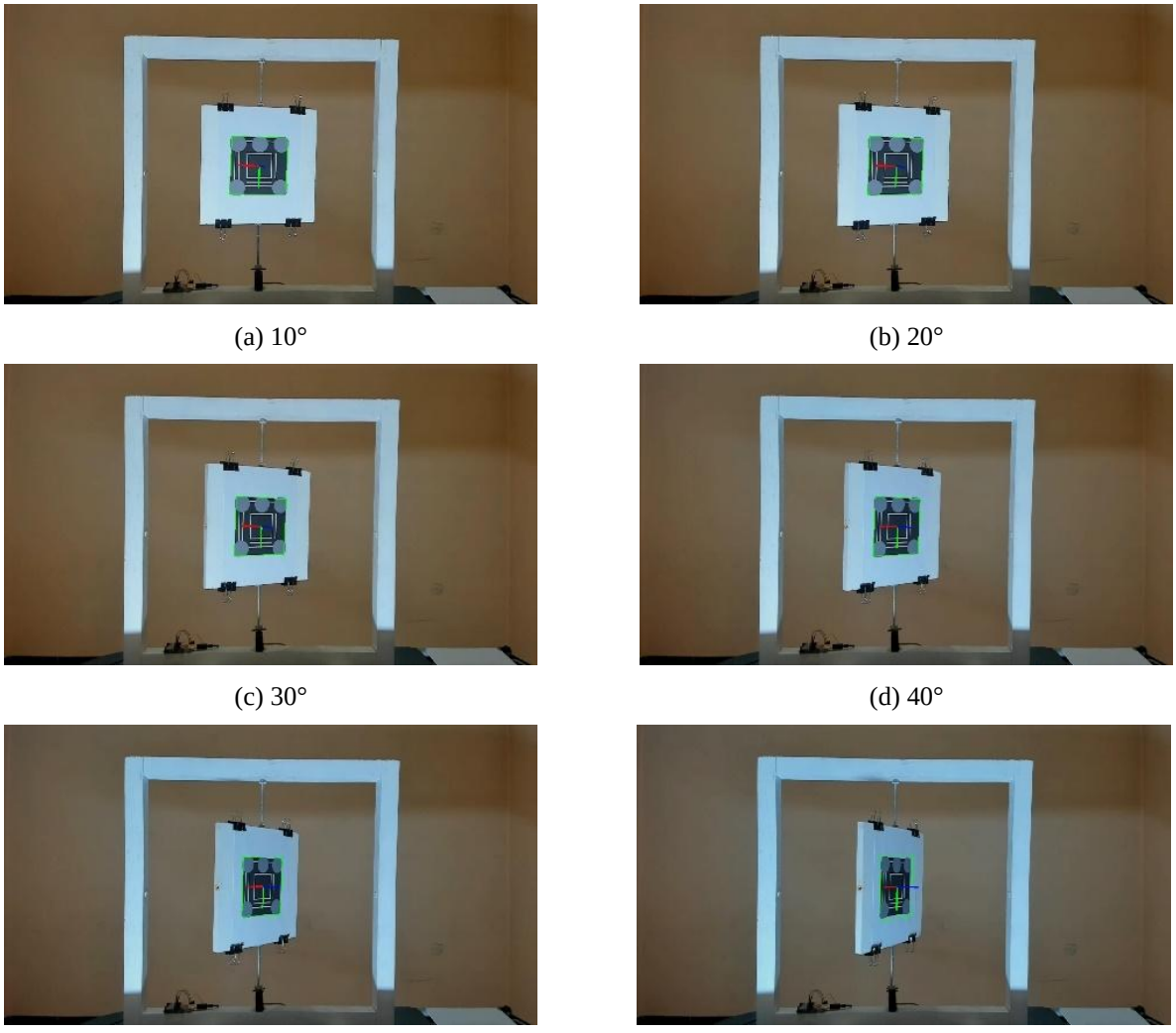
(c) Occluded April marker pose estimation failure



(d) OILU Marker identification and pose estimation is possible even under occlusion

Figure 17: Snapshots from video comparing pose estimation using ArUco, April, and OILU Markers.

To further evaluate the robustness of the proposed OILU marker under occlusion, we conducted tests on real images with varying levels of occlusion and angular rotations. Figure 18 illustrates sample results for the OILU marker with 40% occlusion, where the marker was rotated at angles between 10° and 60°. Despite the challenging conditions, the OILU marker maintained detectable performance across all tested angles, showcasing its superior occlusion resilience. The images clearly demonstrate that the marker remains identifiable even at larger angles like 50° and 60°, where visibility is further reduced due to both occlusion and perspective distortion. Additionally, quantitative results presented in the angular error graph illustrated in Figure 19, confirm the impact of increasing occlusion levels (10%–50%) on pose estimation accuracy. Angular error tends to increase as occlusion levels rise, particularly beyond 30%. However, the proposed OILU marker still performs reliably. These results highlight the OILU marker's robustness in handling substantial occlusions and diverse viewing angles, demonstrating its reliability for pose estimation tasks in challenging real-world environments. The combination of visual examples and quantitative analysis emphasizes the marker's superior performance, particularly when other markers fail to deliver consistent results.



(a) 10° (b) 20°

(c) 30° (d) 40°

(e) 50° (f) 60°

Figure 18:Pose estimation of OILU marker under 40% occlusion at different angles.

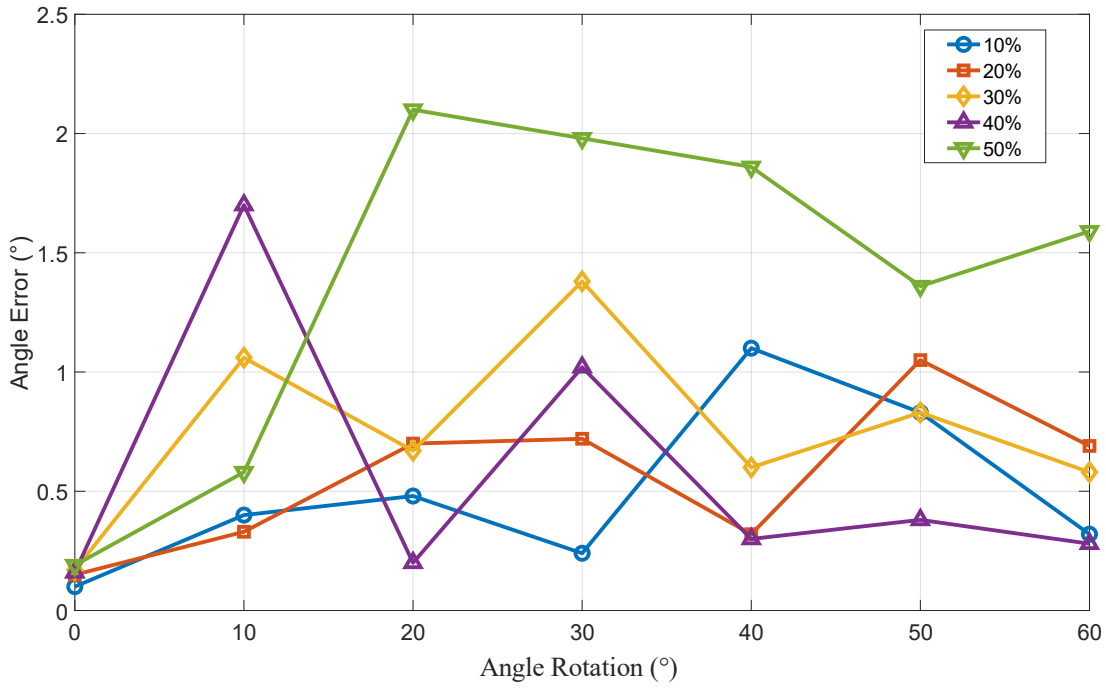


Figure 19: Effect of Occlusion (10%–50%) on Angular Error at Various Rotation Angles.

6.4 Execution time performance evaluation

The identification method described in Section 5.2 has been implemented and compared with the available ArUco [7] and April [8] tools using a laptop equipped with a 2.4 GHz Intel Core i7 processor with 16 GB RAM running on Linux. Reported processing times (Table 6) show that the proposed OILU system requires less processing time than the ArUco and April systems at all processed image resolutions. The gap between the different approaches is more important when dealing with multiple markers within the camera front of view (see Figure 20).

Such results confirm that the improved OILU solution outperforms state-of-the-art solutions in terms of rapid identification. The remaining challenge is developing a full hardware solution embedding the OILU marker identification process within a single System-on-Chip (SoC) device [31], ensuring fluid identification for highly constrained SLAM applications.

Table 6: Processing times with one marker (ms).

Image Resolution	OILU Tag	ArUco (36h12)	ArUco (16h3)	April (25h9)	April (36h10)
Low Resolution (640x480)	6,89	12,91	12,29	12,92	14,95
Medium Resolution (800x600)	8,51	12,98	12,99	13,60	15,44
High Resolution (1920x1080)	17,20	21,13	18,14	18,88	22,15

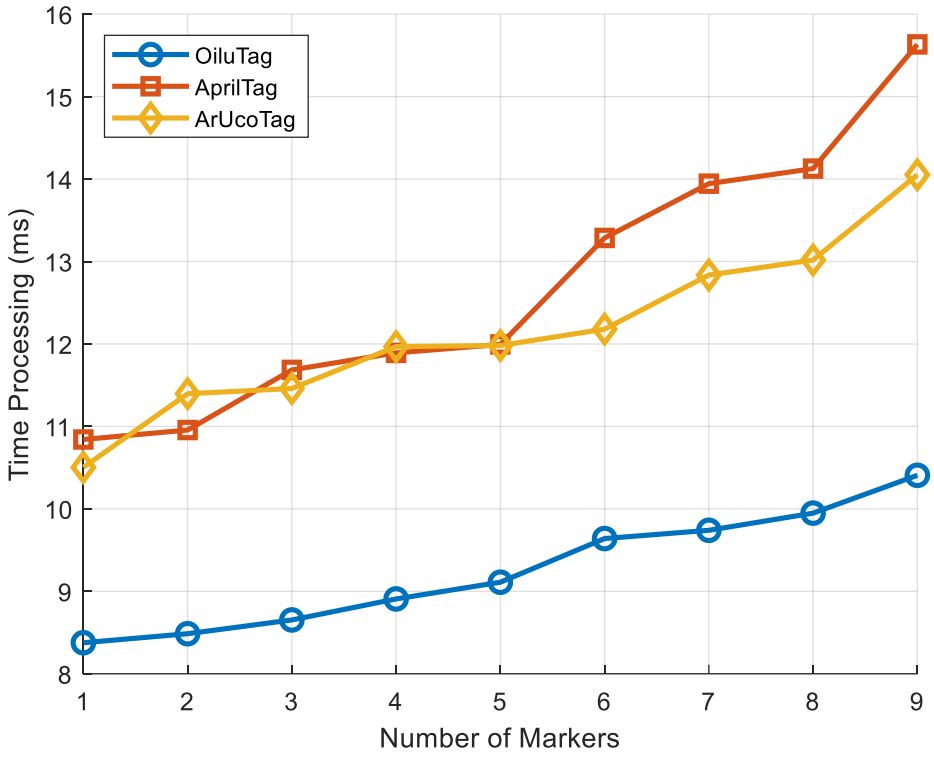


Figure 20: Processing time based on the number of markers within the camera front of view. Image resolution (1920×1080) and markers size (5cm×5cm).

7 Conclusion

An improved OILU marker system design is proposed for an accurate detection and identification scheme. Two approaches have been validated. The first one (based on cumulative histogram analysis) includes homography to process standard OILU markers. A second homography-less identification scheme is proposed to further improve marker detection and identification performances. The last involves enclosing the embedded identifier within two nested square-like quadrilaterals, allowing robust marker detection and identification even under challenging occlusion distortions.

Compared with the main state-of-the-art markers, the proposed approach presents approximately similar detection and identification results but with fewer computational resources and, consequently, less processing time. The suggested marker design, characterized by its consistent line-based pyramidal structure, surpasses standard markers in handling difficult occlusion distortions. Particular attention is paid to the possibility of identifying and estimating the pose of these markers, even if the external marker's corners are occluded.

At this stage, the developed marker does not integrate a corrector code. An improved OILU marker design, including a CRC code, is under development. It allows retaining or rejecting marker identifiers without significantly affecting the marker's codification capacities. Overall, the primary aim of this work is to underscore the potential benefits of employing uniform line-based 2D markers as a viable alternative to established state-of-the-art markers. Future work will extend the application of OILU markers to visual simultaneous localization and mapping (SLAM) projects, where markers are used to embed various environmental and orientation information exploited by uncrewed aerial vehicles (UAV) for accurate navigation and landing.

Acknowledgments

The authors would like to thank the Private Institute of Optics BBA for the support provided for the production and publication of this work.

References

[1] Alam, M.S., Gullu, A.I. & Gunes, A. Fiducial Markers and Particle Filter Based Localization and Navigation Framework for an Autonomous Mobile Robot. SN COMPUT. SCI. 5, 748 (2024) https://doi.org/10.1007/s42979-024-03090-y

[2] Romero-Ramire, F. J., Munoz-Salinas, R., Medina-Carnicer, R. Fractal Markers: a new approach for long-range marker pose estimation under occlusion. IEEE Access, (2019), 7, pp. 169908–

169919.
http://dx.doi.org/10.1109/ACCESS.2019.2951204

[3] Zhenglong, G., Qiang, F., & Quan, Q. Pose estimation for multicopters based on monocular vision and AprilTag, 37th Chinese Control Conference (CCC), 2018, pp. 4717–4722. http://dx.doi.org/10.23919/ChiCC.2018.8483685

[4] Al-Dhaimesh, Sadeer Hel, and Nooriati Taib. "A review: investigation of augmented reality–BIM benefits in design process in AEC industry." Informatica 47.5 (2023). https://doi.org/10.31449/inf.v47i5.4671

[5] Sani, M. F., & Karimian, G. Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. International Conference on Computer and Drone Applications (IConDA), 2017, 102–107. http://dx.doi.org/10.1109/ICONDA.2017.8270408

[6] Sarmadi, H., Muñoz-Salinas, R., M. Álvaro, B., Luna, A., Medina-Carnicer, R. 3D Reconstruction and alignment by consumer RGB-D sensors and fiducial planar markers for patient positioning in radiation therapy, Computer Methods and Programs in Biomedicine,Volume 180,2019,105004, https://doi.org/10.1016/j.cmpb.2019.105004

[7] Olson, E. AprilTag: A robust and flexible visual fiducial system. IEEE International Conference on Robotics and Automation, (2011), pp. 3400–3407. (2011) http://dx.doi.org/10.1109/ICRA.2011.5979561

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

[9] Rhijn, A., Jurriaan, M. Optical Tracking using Line Pencil Fiducials. (2004), 10.2312/EGVE/EGVE04/035-044.

[10] Chahir, Y., Mostefai, M., & Saidat, H. New Efficient Visual OILU Marker, The 25th International Conference on Image Processing Computer Vision, & Pattern Recognition (IPCV 2021), Book of Abstracts, 138, ISBN # 1-60132-514-2, (2021).

[11] Adalsteinsson, D., Sethian, J. A Fast Level Set Method For Propagating Interfaces, Comp Phys., (1995), Vol. 118, pp. 269-277. doi:10.1006/jcph.1995.1098

[12] Kato, H., & Billinghurst, M. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99),(1999), pp. 85–94. http://dx.doi.org/10.1109/IWAR.1999.803809

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

[14] Fiala, M. Comparing ARTag and ARToolkit Plus fiducial marker systems. IEEE International Workshop on Haptic Audio Visual Environments and Their Applications, (2005). 6--pp. http://dx.doi.org/10.1109/HAVE.2005.1545669

[15] Yu, G., & Hu, Y., & Dai, J. TopoTag: A Robust and Scalable Topological Fiducial Marker System. IEEE Transactions on Visualization and Computer Graphics. (2020). http://dx.doi.org/10.48550/arXiv.1908.01450

[16] DeGol, J., Bretl, T., & Hoiem, D. Chromatag: A colored marker and fast detection algorithm. Proceedings of the IEEE International Conference on Computer Vision, (2017), pp. 1472–1481. https://doi.org/10.1109/ICCV.2017.16

[17] Calvet, L., Gurdjos, P., Griwodz, C., Gasparini, S. Detection and accurate localization of circular fiducials under highly challenging conditions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2016), 562–570. http://dx.doi.org/10.1109/CVPR.2016.67

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

[19] Bergamasco, F., Albarelli, A., Rodolà, E., Torsello, A. RUNE-Tag: A high accuracy fiducial marker with strong occlusion resilience. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (2011), pp.113 - 120. http://dx.doi.org/10.1109/CVPR.2011.5995544

[20] Birdal, T., Dobryden, I., Ilic, S. X-Tag: A Fiducial Tag for Flexible and Accurate Bundle Adjustment. (2016), pp. 556-564. http://dx.doi.org/10.1109/3DV.2016.65

[21] Burak, B., Cihan, T., Cuneyt, A. STag: A stable fiducial marker system, Image and Vision Computing, Vol 89, 2019, pp.158-169. https://doi.org/10.1016/j.imavis.2019.06.007

[22] Shingo, K., Hashimoto, K. Homography Estimation Using Marker Projection Control: A Case of Calibration-Free Projection Mapping, IFAC-Papers On Line, Vol 56, Issue 2, 2023, pp.

2951-2956.
http://dx.doi.org/10.1016/j.ifacol.2023.10.1418

[23] Gonzalez, R. C., Woods, R. E. Digital image processing. Pearson Education limited, 4th Edition. (2017). https://doi.org/10.1117/1.3115362

[24] Suzuki, S., Be, K. , Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1), (1985), 32–46. https://doi.org/10.1016/0734-189X(85)90016-7

[25] Douglas, D. H., & Peucker, T. K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: International Journal for Geographic Information and Geovisualization, 10(2), (1973), pp. 112–122. http://dx.doi.org/10.3138/FM57-6770-U75U-7727

[26] Li, Y., Zhu, S., Yu, Y., & Wang, Z. An improved graph-based visual localization system for indoor mobile robot using newly designed markers. International Journal of Advanced Robotic Systems, 15(2), (2018). https://doi.org/10.1177/1729881418769191

[27] Otsu, N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), (1979), pp. 62–66. http://dx.doi.org/10.1109/21.35351

[28] Xuancen, L., Shifeng, Z., Jiayi, T., Longbin, L. An Onboard Vision-Based System for Autonomous Landing of a Low-Cost Quadrotor on a Novel Landing Pad, Sensors, (2019), 19, 4703. https://doi.org/10.3390/s19214703

[29] Cong, Yingzi. "Image stitching technology for police drones using an improved image registration method incorporating ORB algorithm." *Informatica* 48.2 (2024). http://dx.doi.org/ 10.31449/inf.v48i2.5877

[30] Hartley, R., Zisserman, A. Multiple View Geomerty in Computer Vision. Cambridge University Press, second edition, 2003. https://doi.org/10.1017/s0263574700223217

[31] Vasileios, L., Panagiotis Minaidis, P., Lentaris, G., Dimitrios, S. Accelerating AI and Computer Vision for Satellite Pose Estimation on the Intel Myriad X Embedded SoC, Microprocessors and Microsystems, Volume 103, (2023). https://doi.org/10.1016/j.micpro.2023.104947