

# An efficient, dense and long-range marker system for the guidance of the visually impaired

Juan M. Sáez<sup>1</sup>, Miguel A. Lozano<sup>1</sup>, Francisco Escolano<sup>1</sup>, and  
Javier Pita<sup>2</sup>

<sup>1</sup>Department of Computer Science and Artificial Intelligence,  
University of Alicante

<sup>2</sup>Nuevos Sistemas Tecnológicos S.L.

February 26, 2021

## Abstract

In this paper, we address the problem of making a mobile/smartphone camera sensitive to distant fiducial markers. To this end, we carefully design a novel visual marker that is both dense and readable from large distances. The main novelty of the proposed marker is the combination of a quaternary color-based coding system with robust methods for reading the color patterns included in each frame once it is detected. These patterns include a CRC whose length grows linearly whereas that of the message grows quadratically. Our experiments show that the proposed bundle *marker-vision algorithm* outperforms the alternatives in terms of distance and angle and also that it is very robust to changes in lighting conditions, thus making it a good intelligent system for guiding people with visual impairments in their day to day use of public transportation systems.

## 1 Introduction

An artificial visual marker is an element that is intentionally introduced in the scene and contains information related to the environment (e.g. a description of the scene, or signposting). Such information is intended to be captured and recognized by the digital camera integrated in a mobile device (e.g. a smartphone). Unlike other kind of markers such as barcodes, whose information is mainly displayed unidimensionally, visual markers display their information in a bidimensional layout. Attending to their design, there are two families of visual markers: dense markers and long-range markers.

**Dense visual markers** Their layout usually consists of a large number of visual elements, to enable the potential coding of a large amount of information. These codes are readable if (i) we have a sharp image, (ii) the camera is close enough (about  $20 - 50cm$ ) and (iii) we are working under good lighting conditions. These requirements prevent dense markers from being detected in certain situations, namely from large distances, while the device is moving (due to the motion blur effect), or when lighting conditions are not ideal.

In this group, we find several well-known types of markers, such as QR codes or the Datamatrix system [13]. These codes are usually dichromatic (consisting of black and white elements), but there are some exceptions such as HCCB (High Capacity Color Barcodes), a bidimensional color-based marker model developed by Microsoft [10]. In HCCB, color (leading to multi-chromatic elements) makes the code denser (a larger amount of putative codes is available). However, the resulting marker is difficult to read (it must be image centered) and this rules out intelligent detection.

**Long-range visual markers** They are designed to improve the detection process: they can be detected from a large distance (typically within a range of about  $5 - 7m$ ) and in different lighting conditions. This kind of markers is mainly applied to Augmented Reality, although they are also used in other context such as Mobile Robotics. These markers usually store much less information than the dense ones, because the more information they include, the more complex is the marker's geometry, thus affecting the recognition process.

The ARToolkit [7] [6] is the most extended long-range marker and it is designed for Augmented Reality. It consists of a black frame over a white background, containing a black and white pattern that encloses the marker. This allows to detect whether a black frame is a marker or not, but it cannot distinguish between different markers. Therefore, ARTs are limited to applications that use a single marker. This family of markers has evolved to systems that are able to include information in the markers, such as AprilTags [9] and Aruco [5]. Their marker localization system is that of ART (black frame over white background) and they basically differ in their encodings. The AprilTags marker, for instance, uses a grid of  $6 \times 6$  dichromatic cells. These 36 elements represent a binary code that contains both the message and a code to detect and correct reading errors, which allows (discarding symmetries and ambiguities) to generate 500 different codes. On the other hand, Aruco uses a smaller grid ( $5 \times 5$ ). It applies a variation of Hamming codes that provides 10 bits of information, allowing to generate 1024 different codes. The rest of the elements in the code are used to detect and correct errors.

In ChromaTags [3], colored markers are introduced in order to improve detection, providing a fast marker detection algorithm, but the density of these markers is still low (each cell encodes 1 bit). For instance,  $4 \times 4$  (16H5),  $5 \times 5$  (25H9) and  $6 \times 6$  (36H11) families of ChromaTags provides 30, 35 and 587 different combinations respectively.

The review of existing visual markers leads us to identify a gap: there is not



Figure 1: Examples of applications of the visual markers for people with visual impairments: Product labeling (left) and signposting in an urban context (right).

an intermediate system which is both dense and readable from large distances.

**Motivation** From the very beginning our aim has been to increase the density of the code with a minimal loss or even an improvement of detection. This is motivated by the need of tagging large environments (metro/ underground stations, bus-lines and the like reaching the scale of a city). Consequently, we propose a hybrid (both dense and long-range) visual marker which can be read by mobile devices designed for the guidance of people with visual impairments. Current visual marker systems have demonstrated to be very useful, allowing to provide information about the environment through a device camera, but they present important limitations in the context of accessibility for people with visual impairments: (i) dense markers need a proper camera focusing that is difficult to achieve by a blind person and (ii) long-range markers can only store a very limited amount of information, while we aim at labelling product prices in a supermarket, or directions in an airport.

Herein, we propose a novel marker system to reach our goals. Compared to some tagging/localization solutions, our system is low-cost both for the user (using smartphone cameras) and for the installer (cheaper than bluetooth beacons and RFID tags). In addition, some localization technologies either do not give us both our exact localization and orientation (beacons) or they are not applicable in indoor environments (GPS). Finally, other low-cost tagging solutions (such as QR and barcodes) do not work at long-range distances.

For this reason, we propose a marker system that combines the advantages of each family, i.e. the capacity of both storing a sufficient amount of information and being detected in a robust and efficient way (with different lighting conditions, in movement and from a large distance). This kind of markers could be applied to contexts in which both range and density play a key role (see Figure 1). We find previous proposals of artificial markers systems aimed at using mobile devices in the context of visually impaired people assistance. In [2] we find a kind of marker that is designed exclusively for this context. It is a unique marker model which does not store information, thus limiting its applications.

Nevertheless, it has inspired our proposal. In [8] we find a marker system for people with visual impairments that combines both long-range and dense features. In this case, the marker detection could be performed from a long range but the marker reading must be performed very close to the target, because the tagged information is given by a dense QR code.

To the best of our knowledge, there are few tests of long-range markers for people with visual impairments. One of these works [4] was carried by our colleagues of SKERI (Smith-Kettlewell Eye Research) in San Francisco. They use Aruco [5] for localization of blind users. This experience reveals the potential usefulness of the long-range artificial markers in applications for these community. Consequently, our goal is to provide a user experience close to long-range systems, but with enough density to cover real life applications.

The rest of the paper is organized as follows. Firstly, in Section 2 we present the design of the layout of our marker proposal, combining both dense and long-range properties. Next, the algorithm to detect these markers is detailed in Section 3 and in Section 4 we show some experimental results from testing our marker system both in the laboratory and in real environments. Finally, in Section 5 we draw some conclusions about the obtained results.

## 2 Marker design

Our marker presents some structural similarities with long-range markers such as AprilTags [9] and Aruco [5]. In this regard, they share the same localization method (black frame over a white background), which is widely used within the family of long-range markers. This frame contains a data grid consisting of  $N \times N$  square cells.

The geometric elements of the proposed marker are shown in Figure 2. The side length  $l$  of each cell determines the width of both black and white location frames and thus the total area of the marker. The main difference between the proposed marker and the previous ones is that the data grid consists of 4-state cells (4 colors) instead of 2-state ones (black and white). Given a grid size  $N$ , our marker model *double-exponentiates* the density of the code (i.e. given a code with  $n$  elements, our marker yields  $4^n$  combinations, instead of the  $2^n$  combinations yielded by a 2-state one). This intrinsic feature does not bound the resolution of the elements of the marker. On the contrary, we can pack less (and yet larger) elements in the same area, to make the marker more readable from a large distance (long-range marker) and still hold a notably high density with respect to the binary case.

Similarly to binary systems, which assign a color to each state (black=0, white=1), in our quaternary system each state is represented by four colors. The choice of these specific colors is open, since the design of the marker includes the palette in the marker itself. Figure 3 shows different representations of the same  $5 \times 5$  marker.

However, it is convenient to select colors that are distinguishable between them in order to avoid ambiguities and therefore simplify (and make more ro-

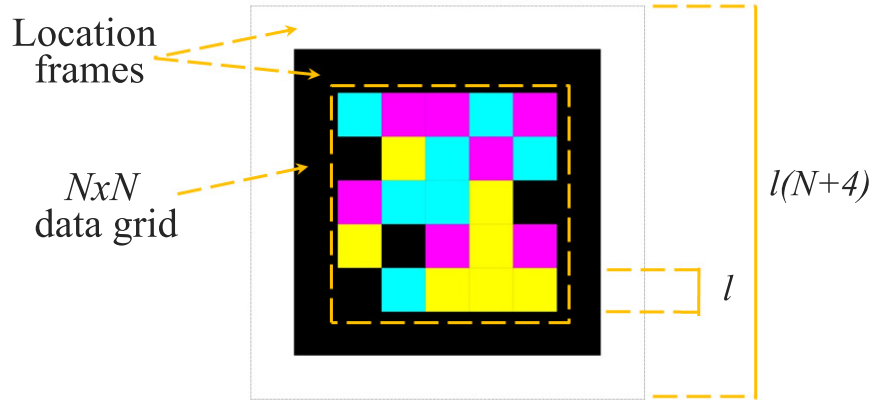


Figure 2: Layout of the proposed marker: location frames (black over white) and a data grid of  $N \times N$  elements ( $5 \times 5$  in this case). Both the square data grid elements and black and white frames share the same width  $l$ . Thus, the area of the whole marker is  $(l(N + 4))^2$ .

bust) the recognition process. In this regard, the combination CMYK (Cyan, Magenta, Yellow, black) is recommended, because it corresponds to the most distant corners in the cube defined by the RGB color space (see Figure 4-left). Within this space, we also find the combination RGBW (Red, Green, Blue, White), that presents the same ability to distinguish the different colors. However, the triplet CMY doubles the luminance of triplet RGB, making the former one more visible in low-lighting conditions. From now on, CMYK will be used in all the graphic examples, although our proposal is not limited to this combination.

The identification of specific colors in a digital image is a classic problem in the field of Computer Vision, which does not have a definitive solution yet. In

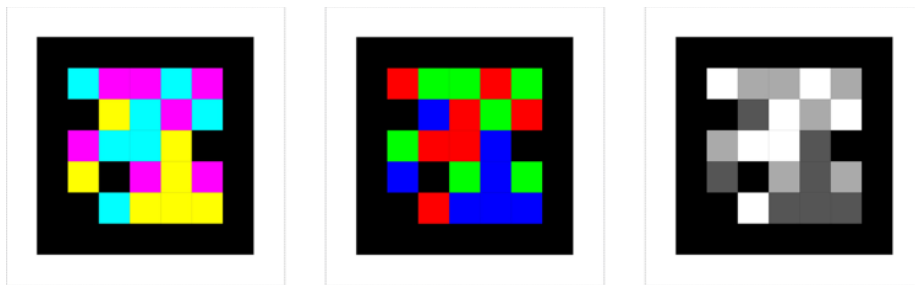


Figure 3: Representations of a single  $5 \times 5$  marker with different color palettes: Cyan/Magenta/Yellow/black (left), Red/Green/Blue/black (center) and four gray tones (right).

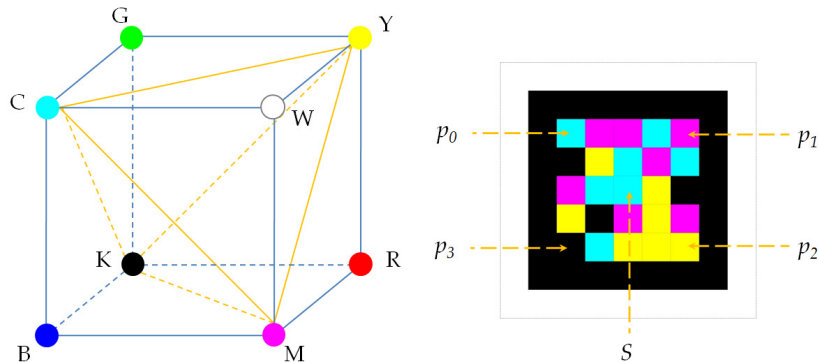


Figure 4: Left: RGB color space. CMYK combination contains the four colors more distant between them and thus more distinguishable. The symmetric combination RGBW presents the same feature, but it is darker. Right: Embedding the palette  $P = [p_0, p_1, p_2, p_3]$  in the marker itself. The palette is located in the corners of the data grid in clockwise order and using as a reference the element with lowest luminance (black element in CMYK case), that is assigned to  $p_3$ . The marker type element  $S$  is located in the central cell of the data grid and determines its size  $N$ .

our case, in the process that takes place since the marker is generated until it is captured by the device camera, its colors may be degraded, making them more difficult to be recognized. In the printing process (generation) we have to take into account different factors (e.g. paper type or printer state), that may degrade the colors. In the next step (image capture), there are much more factors that affect the colors (e.g. lighting, camera setup and lens quality). This causes that, to a greater or a lesser extent, the observed color presents notable differences with respect to the original one. In our case, we must also consider the degradation of the marker colors caused by ambient factors over time (humidity, exposition to sun, oxidation, etc.). For these reasons, the approach consisting of looking for absolute colors in the image is discarded. Instead, we provide robustness by including a palette in the marker itself. The palette consists of a set of four (reference) colors that will be used in the marker  $P = [p_0, p_1, p_2, p_3]$ . This palette is embedded in the marker itself, in the four corner elements of the data grid (see Figure 4-right). It is introduced in clockwise order and its reference (last element  $p_3$ ) is given by the darkest (lowest luminance) element in the palette. This reference makes the marker invariant to rotation transforms. The inclusion of the palette solves the problem of the color identification in the recognition method (it will be detailed in Section 3). Once we obtain a reference for each color, a proximity criterion may be used to label the cells of the data grid. Thus, the above-mentioned color degradation problems are avoided, because the color of each cell is not compared to an ideal color. It is expected that the palette presents a similar degradation degree than the color cells it is

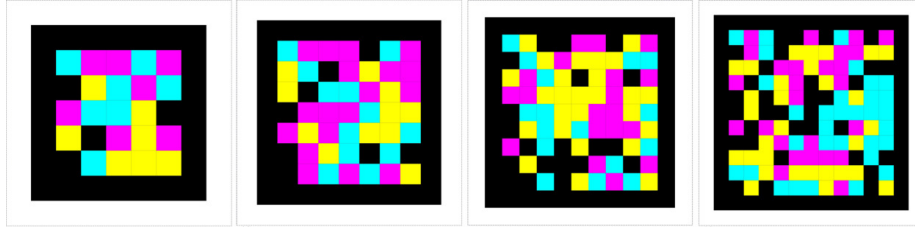


Figure 5: Examples of the different supported marker sizes. From left to right:  $5 \times 5$ ,  $7 \times 7$ ,  $9 \times 9$ ,  $11 \times 11$ . For the CMYK palette, the central data grid cell color represents the following sizes: cyan for  $5 \times 5$ , magenta for  $7 \times 7$ , yellow for  $9 \times 9$  and black for  $11 \times 11$ .

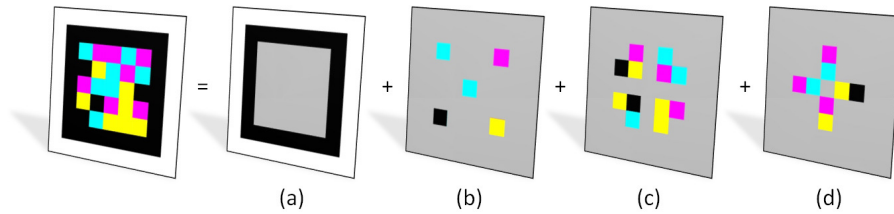


Figure 6: Elements of the marker: (a) location frames, (b) palette and size, (c) message and (d) CRC.

compared with.

In addition, the proposed marker is defined for four data grid sizes:  $N \in \{5, 7, 9, 11\}$ . We can choose a given size depending on the application we are aiming the markers to. Reading the marker becomes more difficult as we increase the marker size (it is more affected by distance, movement and other factors). We consider that the system loses its long-range capabilities with resolutions higher than  $11 \times 11$ . This fact motivates the design decision of supporting these sizes. The color of the central data grid cell encodes the size of this marker. Note that all the supported sizes are odd, so it always exists a unique central cell. This central cell may take every color in our palette  $S \in \{p_0, p_1, p_2, p_3\}$  (like the others cells in the grid) and these colors correspond to data grid sizes  $\{5, 7, 9, 11\}$  respectively (see Figure 5).

The remaining cells in the grid are devoted to the message and the Cyclic Redundancy Check (CRC) [11]. The CRC is computed from the message and it is included in the marker for later checking. In the recognition step, the message is extracted, the CRC is computed from the message and it is compared to the CRC in the marker in order to validate its integrity. This procedure is widely applied in the field of communications systems.

The CRC code is included in the central row and column (except the central cell, that is used to encode the marker size). The message is stored in the rest of free elements (see Figure 6), that is, the ones that are not dedicated to encode

Table 1: Message length, CRC length and generator polynomial for each marker size.

$N \times N$	Message Size	CRC Size	CRC Polynomial
$5 \times 5$	24 bit	16 bit	CRC-16-CDMA2000
$7 \times 7$	64 bit	24 bit	CRC-24-Radix-64
$9 \times 9$	120 bit	32 bit	CRC-32Q
$11 \times 11$	192 bit	40 bit	CRC-40-GSM

the palette, marker size and CRC. Given that each cell allocates 2 bits (four values), a  $N \times N$  marker contains  $2N^2 - 4N - 6$  bits of data and  $4N - 4$  bits of CRC. We can see that the message and CRC length grow quadratically and linearly respectively. Table 1 shows, for each marker size, the message and CRC length and the generator polynomial used to compute the CRC. We have used standard generator polynomials according to each length, whose effectiveness has been widely proved.

The inclusion of the color in the marker design has a notable impact in the message density. The Aruco system with  $5 \times 5$  markers provides 1024 combinations, AprilTags with  $6 \times 6$  markers leads to 500 combinations and ChromaTags with  $5 \times 5$  markers includes only 35 combinations (given that in this case color is used to improve marker detection instead of density), whereas the smaller version of our proposal ( $5 \times 5$ ) is able to store a 24-bit message, that is, 16.777.216 combinations.

### 3 Marker detection

Once we have detailed the design of the marker layout, in this section we describe the procedure for detecting this kind of markers within an image.

Given a color digital image  $I(x, y)$ , which has been captured by the device camera and a marker size  $N \in \{5, 7, 9, 11\}$ , the set of markers  $T$  that this image contains can be obtained as follows.

**Marker location** The first step consists of locating the marker (or markers, if there are many of them) in the image. We have seen in Section 2 that the marker has a black frame over a white background, which is used for this purpose. This is a pervasive element in long-range markers due to the high contrast given by black over white. Consequently, in this first step, we apply a frame detection algorithm to detect a set of candidates. There are many algorithms that solve this problem (in fact, different long-range marker proposals, such as ARToolkit [7] [6], AprilTags [9] and Aruco [5], include their own frame location algorithm). In our case, we recommend the algorithm included in Chilitags library [1], but any other frame detection method can be a good alternative. Within the above-mentioned alternative methods, AprilTags stands out for its robustness, at the expense of its notably higher computational cost.





Figure 7: Original image (left). Result of the frame detection algorithm (center).  $N \times N$  grid obtained from frame coordinates.

The result of marker location for the image  $I(x, y)$  is a set of frames  $M = \{m_0, m_1, \dots, m_{(k-1)}\}$ . Each frame is defined by four coordinates  $m_i = (c_0, c_1, c_2, c_3)$  over the image space, that correspond to its outer corners in clockwise order (see Figure 7-center).

**Grid coordinates** Each detected frame  $m_i$  represents a candidate marker. Taking the frame coordinates  $m_i = (c_0, c_1, c_2, c_3)$  as a reference, a bilinear interpolation is applied to obtain a grid  $G_i(x, y)$  with  $N \times N$  equidistant coordinates. Such a grid consists of the coordinates that correspond to the center of the  $N \times N$  data cells in the marker.

**Color samples** For each coordinate in  $G_i$  we obtain the corresponding color in the image  $C_i(x, y) = I(G_i(x, y))$ . Taking into account that  $G_i$  is given in real coordinates and the image in discrete coordinates, such a color is obtained by a 4-neighbors bilinear interpolation.

**Orientation detection** The matrix  $C_i$  contains a reference for the color of the  $N \times N$  data cells in the marker, but its orientation is still unknown (the frame does not contain orientation information). In order to solve this problem, we have to take into account that the marker corners  $\{C_i(0, 0), C_i(N-1, 0), C_i(N-1, N-1), C_i(0, N-1)\}$  contain the palette in clockwise order, having as a reference the darkest element in the last position. Therefore, considering the four possible orientations of  $C_i$ , the oriented matrix  $C_i^*$  is the one in which its element  $C_i^*(0, N-1)$  contains the darkest element (lowest luminance) of the palette.

**Color palette** Having the oriented color matrix  $C_i^*$ , we obtain the color palette from its corners:

```

Algorithm MARKERDETECTION {
  Given an image  $I(x, y)$  and a marker size  $N \in \{5, 7, 9, 11\}$ 
  Extract frames  $M = \{m_0, m_1, \dots, m_{(k-1)}\}$  from  $I(x, y)$ 
  For each frame  $m_i \in M$ :
    Obtain  $N \times N$  grid coordinates  $G_i(x, y)$  from  $m_i$ 
    Obtain  $N \times N$  color samples  $C_i(x, y) = I(G_i(x, y))$ 
    Obtain  $C_i^*$  orienting  $C_i$  with its reference corner.
    Obtain palette  $P_i = [p_0, p_1, p_2, p_3]$  from corners of  $C_i^*$ 
    Label  $E_i(x, y)$  by the nearest neighbor of  $C_i^*(x, y)$  in  $P_i$ 
    If  $E_i(\lfloor N/2 \rfloor, \lfloor N/2 \rfloor) = (N - 5)/2$  then:
      Extract message  $S_i^m$  and stored CRC  $S_i^c$  from  $E_i$ 
      If CRC from  $S_i^m$  matches  $S_i^c$  then add  $S_i^m$  to  $T$ 
  return  $T$ 
}

```

Figure 8: Marker Detection algorithm

$$P_i = [p_0, p_1, p_2, p_3] = [ \begin{array}{l} C_i^*(0, 0), \\ C_i^*(N - 1, 0), \\ C_i^*(N - 1, N - 1), \\ C_i^*(0, N - 1) \end{array} ] \quad (1)$$

**Marker labeling** From the color palette, we label the marker cells of  $C_i^*$ , obtaining a marker labeling matrix  $E_i(x, y)$ , i.e. a matrix in which each element contains the index in the palette corresponding to the color of this cell. This labeling is obtained through a nearest neighbor classification approach, i.e. we assign the index of the palette color closer to the color of the cell:

$$E_i(x, y) = \arg \min_{k \in \{0, 1, 2, 3\}} \|C_i^*(x, y) - p_k\| \quad (2)$$

In order to calculate the Euclidean distance  $\|\cdot\|$  between two colors we recommend to use the CIE 1976 L\*a\*b\* color space, given that this is an isotropic color space (unlike alternative color spaces such as RGB).

**Validate marker size** Once the cells have been labeled, we have to check whether the marker belongs to the family of markers we are looking for or not, i.e. we have to check that the size specified in the central cell matches the size we are looking for:  $E_i(\lfloor N/2 \rfloor, \lfloor N/2 \rfloor) = (N - 5)/2$ . Otherwise, the current frame is discarded (it is not considered as a valid marker) since the coordinates of the central element will be always correct, even if we find markers with non-matching sizes.

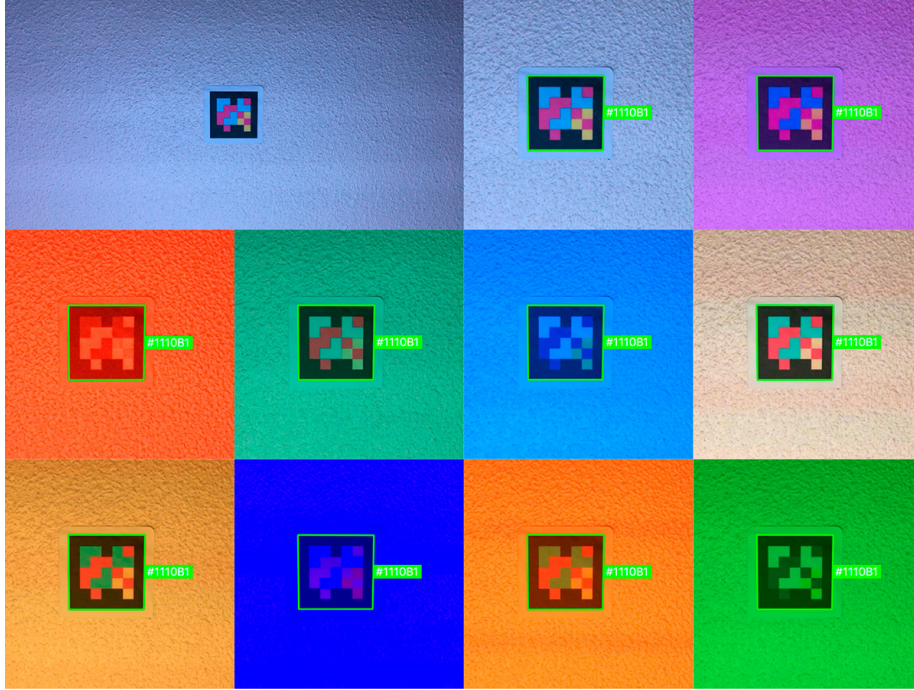


Figure 9: Environmental light color experiment. Original setup with a target on a white wall (top-left) and detection results lighting the scene with 10 different colors.

**Extract information** If the marker size is valid, we proceed to extract the message  $S_i^m$  and the CRC  $S_i^c$ . In order to obtain  $S_i^c$  we take the cells from the center row and column (except the central cell, which determines the marker size) and we build a single  $(4N - 4)$ -bit number composed with the cell labels in binary (labels  $[0, 1, 2, 3]$  correspond to binary codes  $[00, 01, 10, 11]$ ) following the reading order over the matrix (from left to right and top to bottom). The message  $S_i^m$  is composed in the same way, with the cells that do not correspond neither to the palette, nor the CRC, nor the marker size, thus obtaining a  $(2N^2 - 4N - 6)$ -bit binary code.

**Integrity check** Finally, we check the integrity of the message, obtaining the CRC from the message  $S_i^m$  by using the proper polynomial (see Table 1) and comparing it with the stored CRC  $S_i^c$ . If both codes match, the message is considered valid and  $S_i^m$  is added to the set of detected markers  $T$  in the current image  $I(x, y)$ .

In Figure 8 we show an overall scheme of the whole proposed marker detection method.

## 4 Experiments

In this section, we show the strengths and limitations of the proposed method through a set of selected experiments. First, we introduce the laboratory experiments that will evaluate the key elements of the method and finally we conclude with testing the implementations in real environments. All the laboratory experiments are performed with the same device, an iPhone SE from 2016 with a dual-core 1.84GHz processor, with a camera resolution limited to  $1280 \times 720$ . We consider that currently this is a middle-range device. With this device and configuration, the algorithm performs the detection process at 25.7 fps. We have used  $5 \times 5$  tags with the CMYK color palette set.

### 4.1 Environmental light

Environmental light play a key role on any computer vision algorithm, but in color-based methods the light frequency is also decisive. In order to evaluate how the color of the environmental light affects to the detection of our markers, we have illuminated a target with many different light color sources using a color LED projector.

In Figure 9 we show the original setup with the target on a white wall (top-left) and the detection results (marked with green boxes) performed under 10 lights with different frequencies (red, green, blue, orange...). As we can see, the perception of the marker colors varies heavily with respect to the light frequency. Nevertheless, the algorithm is be able to detect the marker in all cases. This is due to the robust labeling step which is performed with respect to the palette. The palette is part of the scene and consequently it is modified by the light in the same way than the rest of colors.

In addition, we have evaluated how the intensity of light affects to the marker detection. In this case, we have performed an experiment with the same setup than the above one, but varying the intensity of light this time. In Figure 10 we show a plot that summarizes 76 frames lighted with different intensities. The intensity of light (vertical axis) is measured through the mean luminosity observed in each frame. The blue part of the plot represents the frames where the target is detected while the red one represents the frames where the method fails. As we can see, the algorithm needs a minimum intensity around 7.71% in order to perform the detection properly. We have included some intermediate frames in order to show the intensity of light represented by this value.

### 4.2 Distance and angle

In order to show that our method is a long-range one, we have performed an experiment in a big hall, where a  $20 \times 20$  cm marker is placed on a wall (Figure 11 bottom left) and observed from 1578 poses, covering an area of  $16 \times 14$  meters. With the aim of providing a comparison of our marker, we have included a  $5 \times 5$  Aruco and a low resolution QR below ours with the same size. We have selected these markers as representatives of long-range and dense families respectively.

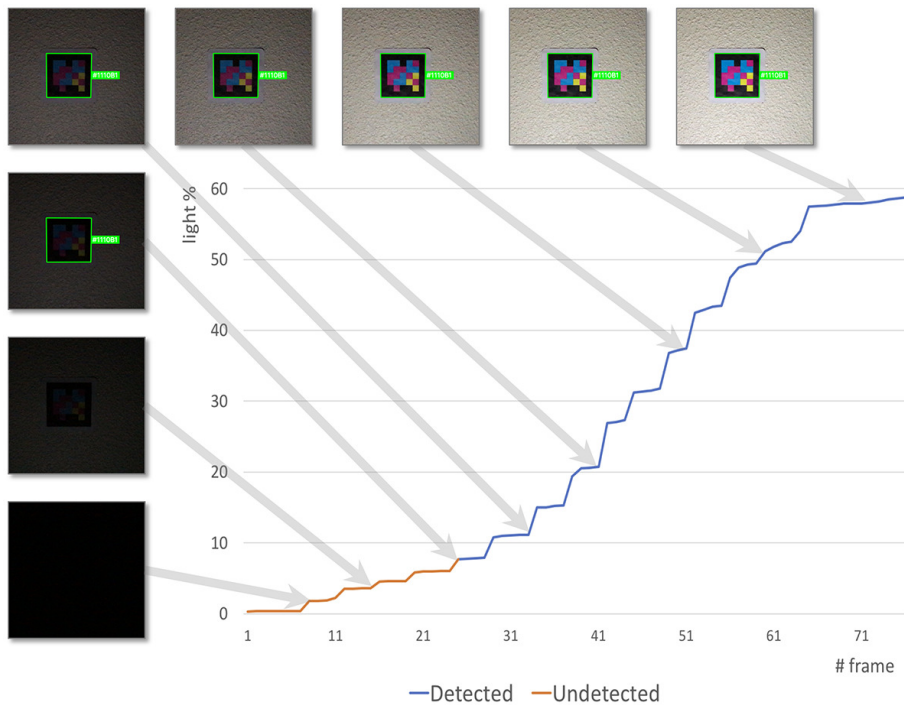


Figure 10: Environmental light intensity experiment. Mean luminosity from 0% to 58% along 76 frames. The target is detected (blue part of the plot) from 7.71% of light.

We ensured that the three markers are visible from each of the 1578 poses. Nevertheless, the detection of each marker depends on both the angle and the distance of the observation. In Figure 11 we summarize the results of the experiment. In the top row we plot the poses from where a marker is detected with our method (left), Aruco (middle) and QR (right). Our method provides the densest results (474 readings, a 29.99% of the total number of observations), that are close to Aruco (378 readings, a 23.90% of the total) and really far from QR (that only obtains 93 readings, a 5.83% of the total). The distributions of readings show a similar shape: the highest density is concentrated in front of the marker and it is getting smaller as the distance or the angle increases. The great difference is in the number of samples and in the size of the distribution, which is much larger for long-range methods.

In the bottom part of Figure 11 we provide another representation of the data in order to evaluate the differences between the methods with respect to the range and to the angle separately. We represent a histogram that shows the number of detections of each method with respect to the distance (middle bottom) and with respect to the angle (bottom right). Here we can see the main difference between long-range and dense methods. On the other hand,

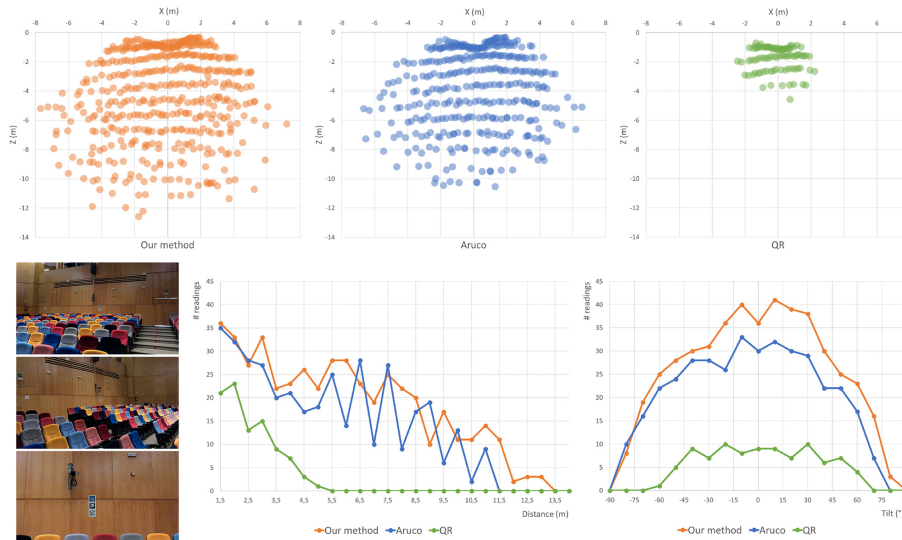


Figure 11: Distance and angle experiment. Marker observed from 1578 poses in a  $16 \times 14\text{m}$  hall, compared with Aruco and QR. Top: 2D distribution of detections with each method. Bottom (from left to right): experimental setup, number of detections of each method vs distance and vs angle.

the slight increment in the number of observations of our method with respect to Aruco is also translated into a slight increment of observation peaks. In terms of range, Aruco is subject to fluctuations while the observation distance increases whereas our marker leads to smoother decrease. Considering the angle, both markers produce few correct readings in extreme conditions, but ours leads faster than Aruco to a larger number of readings as these conditions improve.

This experiment reveals that our method shows the powerful features of the long-range family, while it uses a color based coding that exponentiates the density of the message.

### 4.3 Adverse conditions for detection

In this section we relate the environmental situations where the detection results could be compromised given our experience in real-world implementations. There are some obvious situations that make fail any computer vision based algorithm, like extreme dark or extreme bright environments, very low quality cameras and so on. Here we will focus on environmental conditions that specifically affects to our algorithm and consequently must be taken into account in real implementations.

As we can see in the previous experiments, our algorithm provides good detection results under many lighting conditions. This is thanks to the inclusion of the palette, that offers a reference of each color in the same lighting and



Figure 12: Two examples of sequences affected by a sparkle with sunlight (top) and artificial light sources (bottom).



Figure 13: Example of sequence partially affected by a hard shadow.

capturing conditions than the colors of the code. This also prevents the effects of color degradation, because both palette and code are degraded in the same way. Therefore, it is possible to find situations where the palette and the code are in different light conditions in the same frame. This situation occurs when the tag is partially affected by a sparkle or a hard shadow.

In Figure 12 we show some intermediate frames of two sequences affected by a sparkle from the sunlight (top) and from an artificial light source (bottom). In both situations the light produces a sparkle that illuminates the tag heterogeneously. These aggressive illumination variance provides non-linear differences in the palette and the code colors, which causes the recognition to fail when the sparkle is remarkable (Figure 12 right)

A similar situation could be observed when a tag is partially affected by a hard shadow. In Figure 13 we show two intermediate frames of a sequence when one half of the label is covered by a shadow while the other half is illuminated by the sun. This situation provides hue color differences between the colors in

both parts that sometimes causes the recognition to fail (Figure 13 right).

These situations could be attenuated by different ways. Sparkles are more frequent than partial shadows, and are easily avoided by using matte materials. On the other hand, partial hard shadows are less frequent. In indoor environments with artificial light, these shadows are easily avoided because light and shadows are in fixed positions along the day. Therefore, in outdoor environments these shadows are more difficult to detect, when the light sources are continuously changing. In these environments, we must pay more attention to rigid elements (doors, walls, ...) that project hard shadows, than bushes or trees that project blurry shadows that are less harmful.

#### 4.4 Implementation in real environments

The marker system that we present in this paper is commercialized with the name of VisualTags. The system consists of an adapted mobile application (for Android and iOS) that detects the tags and reads the contents to the user (Figure 14 top left). The application provides the 3D position of the observed tags through a language of sound signals. The contents of the tags are obtained from a remote database which is maintained through a web-based application.

VisualTags was first presented in TifloInnova in November 2017, a key reference in tiflo technology (Figure 14 top right). With the help of the organizers (ONCE, the biggest organization of blind people in Spain) the system was installed to guide the users through the stands of the fair.

In 2018, VisualTags was first implemented in the public transportation system of Barcelona City, Spain (Figure 14 center), in collaboration with TMB (Transports Metropolitans de Barcelona). The first implementation phase included the bus route H12 (396 tags) and the metro stations of the line 9 (833 tags). These tags include useful mobility information such as next arrivals, advertisements or incidences in the language of the user (the application provides the information of each tag translated to the language of the device). Along 2018, some seminars are organized in collaboration with TMB and ONCE in order to show the system to the blind and visually impaired users. In these seminars, we have obtained a very good feedback from the final users, that consider that the system is very useful in their day to day public transport activities. In 2019, the system was extended to fully cover the bus and metro nets of TMB.

The system is also implemented in all the stations of the streetcar in the city of Murcia (Spain), in collaboration with TDM (Tramway of Murcia). We have also deployed VisualTags in the Roman Theater Museum in Cartagena (Spain).

## 5 Conclusions

In this paper, we have examined the integral path of a fiducial marker for people with visual impairments: from the careful design (trade-off between long-range and density) to the implementation (robustness to lighting conditions, rotation invariance) and the deployment of the marker-reader system in real





Figure 14: Implementation in real environments. VisualTags (top-left) was first presented in Tifloinnova'2017 (top-right). Implementations of VisualTags in the public transport of Barcelona (center), Cartagena Roman Theatre Museum (bottom-left) and Murcia Tram (bottom-right)

environments (public transportation systems). Consequently, we have provided the elements of an intelligent system (the bundle marker-mobile/smartphone) which can sense and interpret visual codes in real contexts, thus having a real impact in quality of life of people with visual impairments. This is consistent with our previous developments in the area, such as AOD (Aerial Obstacle Detection) [12].

## Acknowledgements

This work is partially supported by the projects TIN2015-69077-P and RTI2018-096223-B-I00 of the Spanish Government and the grant INFO2016.08.ID+I.0019 from *Instituto de Fomento de la Región de Murcia*. In 2017, VisualTags obtained the XI Vodafone Connecting for Good award by Vodafone Foundation. The methods presented in this paper are protected by the patent P201631625. We also thank James Coughlan from SKERI (Smith-Kettlewell Eye Research Institute, San Francisco, CA) for his insights in the practical use of artificial markers by the visually impaired community.

## References

- [1] Bonnard, Q., Lemaignan, S., Zufferey, G., Mazzei, A., Cuendet, S., Li, N., Özgür, A., Dillenbourg, P.: Chilitags 2: Robust fiducial markers for augmented reality and robotics. (2013). URL <http://chili.epfl.ch/software>
- [2] Coughlan, J., Manduchi, R., Mutsuzaki, M., Shen, H.: Rapid and robust algorithms for detecting colour targets. In: Proceedings of the 10th Congress of the International Colour Association (AIC Colour 05), pp. 959–962 (2005)
- [3] DeGol, J., Bretl, T., Hoiem, D.: Chromatag: A colored marker and fast detection algorithm. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1481–1490 (2017). DOI 10.1109/ICCV.2017.164
- [4] Fusco, G., Coughlan, J.: Indoor localization using computer vision and visual-inertial odometry. In: Computers Helping People with Special Needs (ICCHP) (2018)
- [5] Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* **47**(6), 2280–2292 (2014)
- [6] Kato, H.: Artoolkit: library for vision-based augmented reality. Tech. rep., Institute of Electronics, Information and Communication Engineers Technical Report (2002)
- [7] Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: 2nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99, San Francisco, CA, USA, October 20-21, 1999, pp. 85–94. IEEE Computer Society (1999)
- [8] Lee, C., Chondro, P., Ruan, S., Christen, O., Naroska, E.: Improving mobility for the visually impaired: A wearable indoor positioning system based on visual markers. *IEEE Consumer Electronics Magazine* **7**(3), 12–20 (2018)

- [9] Olson, E.: Apriltag: A robust and flexible visual fiducial system. In: IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011, pp. 3400–3407. IEEE (2011)
- [10] Parikh, D., Jancke, G.: Localization and segmentation of A 2d high capacity color barcode. In: 9th IEEE Workshop on Applications of Computer Vision (WACV 2008), 7-9 January 2008, Copper Mountain, CO, USA, pp. 1–6. IEEE Computer Society (2008)
- [11] Peterson, W.W., Brown, D.T.: Cyclic codes for error detection. Proceedings of the IRE **49**(1), 228–235 (1961). DOI 10.1109/JRPROC.1961.287814
- [12] Sáez, J.M., Escolano, F., Lozano, M.A.: Aerial obstacle detection with 3-d mobile devices. IEEE J. Biomedical and Health Informatics **19**(1), 74–80 (2015)
- [13] Stevenson, R.: Laser marking matrix codes on pcbs. Printed Circuit Design and Manufacture (2005)