

Large Scale Information Marker Coding for Augmented Reality using Graphic Code

Bruno Patrão

*Institute of Systems and Robotics
University of Coimbra - Portugal
bpatrao@isr.uc.pt*

Leandro Cruz

*Institute of Systems and Robotics
University of Coimbra - Portugal
lmvcruz@isr.uc.pt*

Nuno Gonçalves

*Portuguese Mint and Official Printing Office
Institute of Systems and Robotics
University of Coimbra - Portugal
nunogon@deec.uc.pt*

Abstract—In this work, we will present some Graphic Code¹ uses related to large scale information coding applied to Augmented Reality. Machine Readable Codes (MRCs) are broadly used for many reasons. However, they are mostly based on small information (like URLs, id numbers, phone numbers, visit card, etc.). The recently introduced Graphic Code differs from classical MRCs because it is very well integrated with images for aesthetic control (Graphic Code has its aesthetic value than classical MRCs). Furthermore, it is able to code a large amount of information, thus it can also store other kinds of models (like meshes, images, sketches, etc.) for applications that are unusual for classical MRCs. The main advantage of using our approach as an Augmented Reality marker is the possibility of creating generic applications that can read and decode these Graphic Code markers, which might contain 3D models and complex scenes encoded in it. Additionally, the resulting marker has strong aesthetic characteristics associated to it once it is generated from any chosen base image.

Index Terms—Augmented Reality, Machine Readable Codes, Large Information Coding, Graphic Pattern, Steganography.

I. INTRODUCTION

Machine Readable Codes (MRC), such as PDF417 [1], MaxiCode [2], DataMatrix [3] and QR Code [4], got such a huge importance over the decades because their ability to communicate a textual information through an image. More recently, some alternative techniques [5], [6] that embed an image in a QR Code to improve its aesthetic have emerged (we call these techniques by Picture Embedded QR Codes - PEQRC). They transform each non-mandatory pixel from the original QR Code in a set of 3×3 pixels keeping the color of the central, and choosing the color of remaining pixels according to the respective pixel in the embedded image (similar to what is done by dithering). Graphic Code is also inspired by dithering techniques.

Graphic Code is a new Machine-Readable Coding (MRC) method [7] [8]. In Section II we will briefly present it. A full description of this approach is out of the scope of this work, please address to Cruz et. al [7] for further details.

Due to the large coding capacity of graphic code, it can be used for several purposes, including Augmented Reality applications. Patrão et. al. [8] introduced the use of this new coding approach for creation of Augmented Reality applications.

To create these Augmented Reality applications, it is needed some kind of support, such as markers, reference images, or inertial systems. We will strictly focus on markers and new ways of encoding and decoding large quantities of information in them.

The main advantage of using our approach as an Augmented Reality marker is the possibility of creating applications that may read and decode these Graphic Code markers, which can contain complex 3D scenes encoded in it. In this way, changing the marker that the device is pointing, it will change the 3D scene and other assets in it. Graphic Code in this case is more than a typical marker that holds an ID, it therefore is a marker that can convey complex and structured information to the application side. Besides, the resulting marker present strong aesthetic characteristics associated to it as it is generated from any chosen base image.

It is noteworthy that this coding method allows transferring to these types of application much more data than usual MRCs. In Section III we will present a coding capacity evaluation. Likewise, we will show other three kinds of data that may be used for several purposes, such as, Augmented Reality that need a large information encoding: (a) a point cloud, (b) a sketch and (c) an image. They will be presented in Section IV.

II. GRAPHIC CODE

Graphic Code [7] [8] is a Machine Readable Coding (MRC) and so, it can be scanned by any device with a camera. It generates an image in which the pixels are properly organized in 3×3 sets, called cells. Some cell patterns are related to symbols used for coding (the association between a pattern and a symbol is named dictionary). Pixels of the Graphic Code's cells are simultaneously used for aesthetic and coding (they are evaluated all together as a single pattern). It allows the creation of aesthetic codes (like those generated by PEQRC), but also allows each cell to code a symbol (instead of using a set of cells to encode a data message, as it is performed by PEQRC). Thus, this new approach can code much more information than classical MRCs. In the next pages, we will keep focus on how this new coding method is able to code much more data than classical MRC, and what kind of new options emerged with this new possibility.

¹Patent Application: INPI 20171000063377

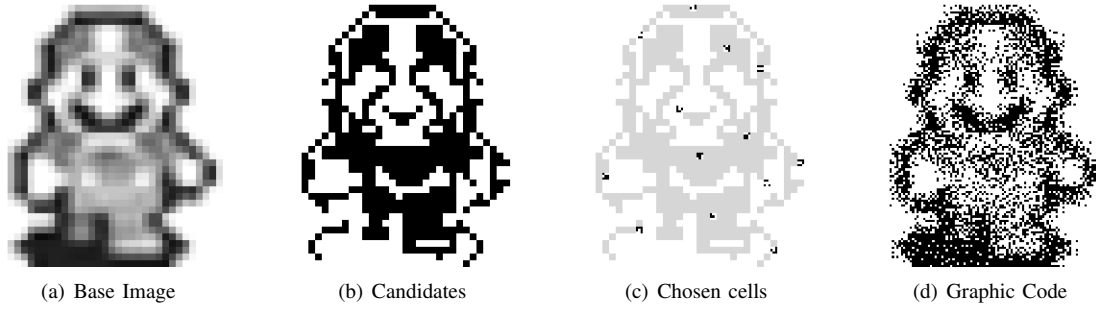


Fig. 1: The tonal coding pipeline starts from the base image (a). Next, it defines the candidate pixels according to the quanta used in the dictionary (black pixels in (b)), then it places the respective pattern of each message symbol (c), and finally it finishes the encoding by replacing the remaining cells.

Quantum	0	1	2	3	4	5	6	7	8	9
Distribution Black/White	9/0	8/1	7/2	6/3	5/4	4/5	3/6	2/7	1/8	0/9
Quantity of Patterns	1	9	36	84	126	126	84	36	9	1
Gray Scale Range	0-26	27-51	52-76	77-102	103-127	128-153	154-178	179-204	205-229	230-255
Quantized Color										
Example of Pattern										

Fig. 2: The quantum grouping of 3×3 black or white pixel cells, and its association to continuous gray levels.

The encoding and decoding processes are symmetrical and both receive as input the same dictionary and grid. Moreover, the encoding input is the message to be encoded, and when necessary, the base image. In the same way, the decoding input is also the graphic code itself.

In order to use alphanumeric dictionaries (with 36 case insensitive symbols, 62 case sensitive symbols, or 75 symbols, combining alphanumeric characters with + - * / # % & @ ' " , .), we can only use colors of the Q3, Q4, Q5 and Q6, since only these have more than 75 patterns, and therefore we can associate each symbol with a different pattern. It therefore follows that, each symbol will have at least four patterns (each referring to one of these).

The main concept of the encoding process is the quantum system shown in Fig. 2. Thus, given a base image (Fig. 1a), each grid cell will be encoded with a pattern which the quantum relates to the color of the respective pixel in it. Since we only use the cells of the quanta Q3, Q4, Q5 and Q6 to associate them with the symbols of the message, we must firstly identify them. We call these cells candidates, which are shown in Fig. 1b as black pixels.

The other steps in the encoding process are similar to what was previously presented. The cells referring to the message symbols are replaced by the respective pattern in the dictionary, and the remaining ones (both unselected candidates and non-candidates) are replaced using non-dictionary patterns (the pattern is chosen according to the quantum of the respective

pixel in the base image, similar to the dithering process). Fig. 1c highlights the candidate cells used for the message symbols, and Fig. 1d shows the final graphic code, with all replacements.

To aid the detection, rectification and decode processes of the marker we use a specific pattern around the coded image. This pattern, as shown in Fig. 4, contains four nested elements: (a) a black rectangle with thickness of 3 pixels, (b) a white rectangle, also with thickness of 3 pixels, (c) a frame with alternating patterns of squares with 3×3 black pixels containing a white pixel in its center, and squares of 3×3 white pixels and a black pixel in its center, and finally (d) a 3×3 black square at the top-left corner adjacent to the alternated squares pattern. Fig. 4 illustrates the alternating squares (left), a full frame (center), and a complete code containing a coded image wrapped by the frame (right).

The detection process searches for these type of patterns in the image and try to rectify the image every time it finds one. The rectification is performed based on points and features of the frame retrieved from the detection process. The purpose of rectification is to achieve an image in which the code appears without distortion (no perspective and no rotation) and with a known scale. Having a calibrated system, the rectified image is achieved by calculating an homography matrix that, when decomposed, represents the transformation between two planes. Last but not least, we calculate the translation and rotation between the marker plane and the camera plane. We also use the 3×3 black square at top-left corner to define the marker orientation.

Once we have the rectified marker image, the decoding process is the inverse of, previous explained encoding process. Using the same order used in encoding, we go through the grid searching for patterns that are in the dictionary. Whenever one of them is found, the respective symbol is chained to the message that is being retrieved.

The full process of encoding and decoding applied to Augmented Reality is presented in Fig. 5.

III. CODING CAPACITY

Current MRCs have widely been used for encoding products identification numbers (product tagging), URLs, business







		Graphic Code	PEQRC	QR Code	PDF417	DataMatrix	MaxiCode
							
	Developer	Ours	Many developers	DENSO Wave	Symbol Technologies	RVSI Acuity CiMatrix	UPS
	Type	Graphic and PE	PE Matrix	Matrix	Stacked barcode	Matrix	Matrix
Data Capacity	Numeric	15288	7089	7089	2710	3116	138
	Alphanumeric	15288	4296	4296	1850	2355	93
	Binary	7644	2953	2953	118	1556	-
	Main features	Enhanced Aesthetic, very large capacity, medium size, high-speed scanning	Aesthetic, large capacity, medium size, high-speed scanning	Large capacity, small size, high-speed scanning	Medium capacity, small size	Medium capacity, small size	High-speed scanning, small size

Fig. 3: Comparison between the graphic code and other machine readable methods.

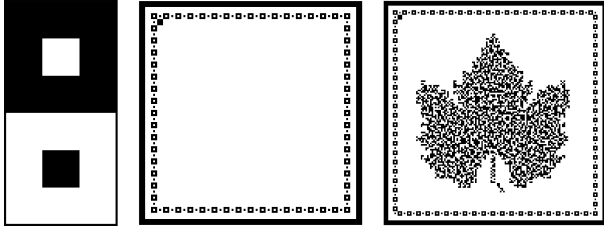


Fig. 4: Frame used to aid the graphic code decoding from a photo

cards, and small texts for general purposes. In all these cases, the encoded message is not too large. This limitation on the coded data dimension refers to the fact that when the code is small then the reconstruction process is more efficient (and classical MRC did not evolve for much bigger models). In the opposite direction of classical MRCs, graphic code needs more resolution for aesthetic improvement. Thus, since its conception, larger coding is an important requirement.

The QR Code standard allows to code at most 177×177 modules, and the largest data capacity (using 8-bits data) for this case is 2,953 bytes [9]. The largest data capacity for a graphic code with the same resolution is 3,481 characters (18% more). This quantity is the number of cells ($3,481 = 59 \times 59$ cells, each one has 3×3 pixels, totalizing the 177×177 image size). However, the proper comparison is between the Graphic Code and Picture-Embedded QR Code (PEQRC). The PEQRC equivalent to the largest QR Code has 531×531 pixels (each module is transformed in 3×3 pixels) and code the same amount of data. The Graphic Code with this resolution can code up to 31,329 characters (10.6x more).

It is important to highlight that, in practical terms, the amount of data is smaller than this quantity, because it depends on the base image (the quantity of pixels related to quanta Q3, Q4, Q5 and Q6). However, it is still much bigger than PEQRC (typically between 5x and 8x more). In addition, it is important to notice that the reconstruction of an image with

531×531 pixels is not such a trivial even assignment. Thus, we are using even smaller images: 180×180 pixels. In this case, we can code 3,600 characters, a PEQRC can code 321, and a QR Code 2,953 (the largest QR Code contains 177×177 modules).

Fig. 3 shows some features of six machine readable methods (including ours), emphasizing that our method can code much more data (these quantities were calculated for codes with the same size of the largest PEQRC). This table is an extension of the one presented by Denso Wave [4].

This increase of data coding allows the use of graphic codes in contexts that are not usual for MRCs nowadays. In Section IV, we will illustrate the use for data transferring in Augmented Reality applications.

IV. APPLICATIONS

All the examples that are being presented use a dictionary containing 97 symbols. Each symbol is mapped to a possible number representation (coordinate or color). The range of this representation is quantized to 97 values, and each possible value is mapped to a symbol. The encoding and decoding applications use this mapping to create the graphic code and recover the model respectively. The Graphic Code was generated using a 60×60 base image (and can code at most 3,600 elements). Furthermore, all these models could be compressed allowing the coding of even larger models.

The pipeline of all examples is the same (Fig. 5). We define a dictionary, choose a base image, define the text to be coded (the model representation), and finally create the Graphic Code. The decoding process begins by capturing a photo of the graphic code, then it decodes the coded information, reconstructs the model, and shows it.

The first example (Fig. 5) is the coding of a point cloud. It has 774 points (totalizing 2,322 numbers). The coordinates were quantized and each value was mapped to a symbol. The decoding application uses the same mapping to recover the point cloud (we also used a reconstruction method to create

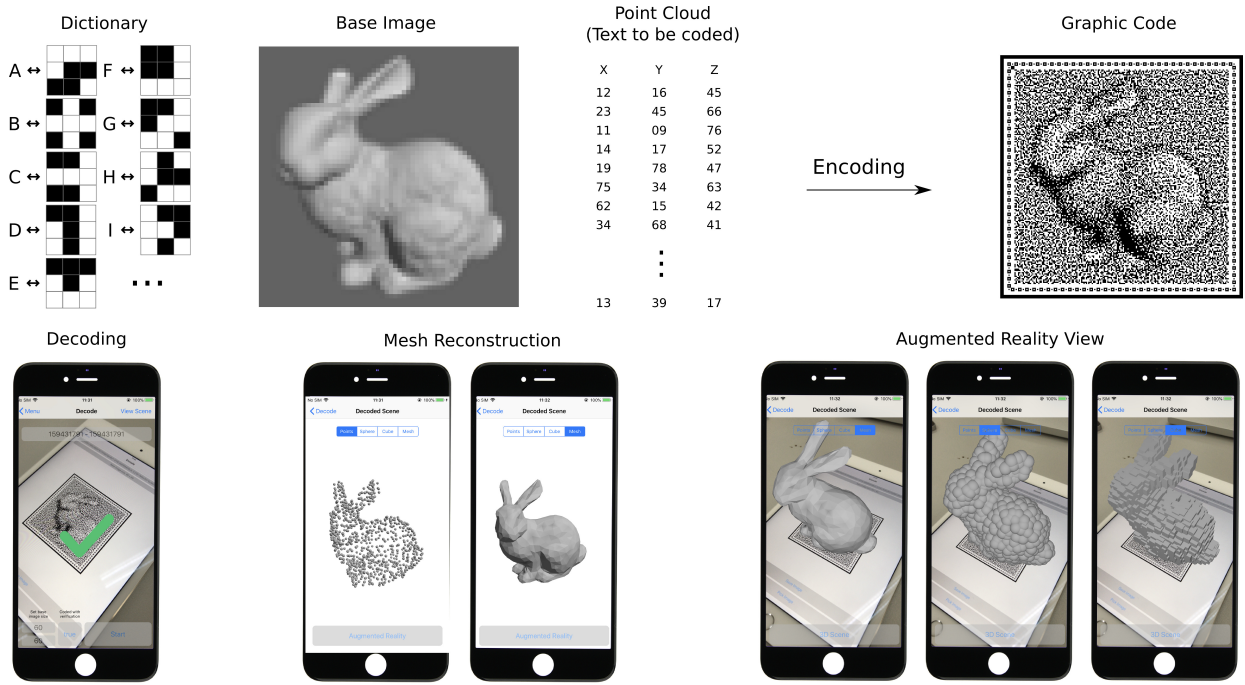


Fig. 5: Full pipeline for encoding and decoding information inside of a marker for augmented reality application.

the mesh from it). The second example (Fig. 6, left) is the coding of an image. We code all pixels of a 40×40 gray scale image into the graphic code. The third example (Fig. 6, right) shows the coding of a sketch (set of 2D points). The sketch is drawn stroke by stroke and the list of 2D points is encoded into a graphic code. Finally, the decoding application reads the code and shows the reconstructed sketch.

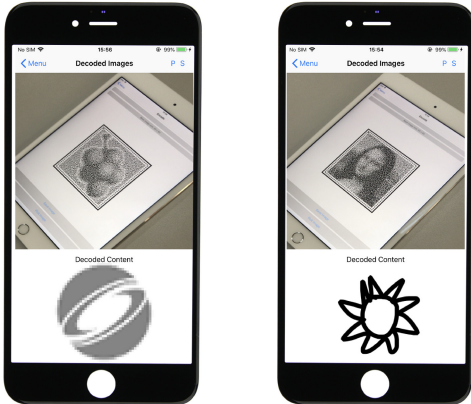


Fig. 6: Two types of models coded into a graphic code: an image (left) and a sketch (right).

V. CONCLUSIONS

All kinds of MRCs encode data that can be provided to any application on demand, without Internet connection, data base access, or even storage on device disk. However, these codes are mostly used for encoding small information like tagging. In this work, we showed that Graphic Code may encode much more data than classical MRCs.

The increase of the encoding capacity allows the creation of applications that handle with larger models (like meshes, curves and images) that are unusual for traditional MRC. Another Graphic Code based application was introduced by Patrão et. al. [8] for creating new Augmented Reality markers. As future work, we intend to code other kinds of data. Some examples include music, descriptions and models for wall panels in museum, among other applications. Another future direction is to combine multiple Graphic Codes to generate even larger models and more complex scenes.

ACKNOWLEDGMENTS

We would like to thank Portuguese Mint and Official Printing Office and University of Coimbra.

REFERENCES

- [1] Y. P. Wang, "Information technology – automatic identification and data capture techniques – pdf417 bar code symbology specification," 2006, ISO standard 15438.
- [2] D. Chandler, E. Batterman, and G. Shah, "Hexagonal, information encoding article, process and system," 1989, uS Patent 4874936.
- [3] "GSI datamatrix guideline," 2017. [Online]. Available: www.gsi.org/docs/barcodes/GSI_DataMatrix_Guideline.pdf
- [4] D. Wave, "Qr code essentials," 2011.
- [5] R. Cox, "Q art code," 2012. [Online]. Available: <http://research.swtch.com/qart>
- [6] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra, "Halftone qr codes," *ACM Transactions on Graphics*, 2013.
- [7] L. Cruz, B. Patrão, and N. Gonçalves, "Halftone Pattern: A New Steganographic Approach," in *EG 2018 - Short Papers*, O. Diamanti and A. Vaxman, Eds. The Eurographics Association, 2018. [Online]. Available: <http://dx.doi.org/10.2312/egs.20181035>
- [8] B. Patrão, L. Cruz, and N. Gonçalves, "An application of a halftone pattern coding in augmented reality," in *SIGGRAPH Asia 2017 Posters*, ser. SA '17. New York, NY, USA: ACM, 2017, pp. 14:1–14:2. [Online]. Available: <http://doi.acm.org/10.1145/3145690.3145705>
- [9] "Qr code," 2017. [Online]. Available: <http://www.qrcode.com>