An Augmented Reality Application Using Graphic Code Markers

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*—This paper lies on presenting applications of the Graphic Code [1] exploiting its large-scale information coding capabilities applied to Augmented Reality. Machine Readable Codes (MRCs) are largely used for many reasons, such as, product tagging or to hold URLs. The recently introduced Graphic Code differs from classical MRCs because it is fully integrated with images for aesthetic control. Furthermore, it is able to code large amount of information and, for that reason, it can store different types of models 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 may contain 3D models and complex scenes encoded in it. Additionally, the resulting marker has strong aesthetic characteristics associated to it.**

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

I. Introduction

Graphic Code [1] [2] is a Machine Readable Coding (MRC) that can be scanned by any device with a camera. The generated code is an image in which the pixels are properly organized in sets of 3×3 pixels, named cells. Some cell patterns are related to symbols and used for coding, the association between a pattern and a symbol is called 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, but also to code a symbol or a set of symbols in each cell. Thus, this new approach can code much more information than classical MRCs [1]. In this work, we will take advantage from this new coding method data capacity and use it to create new markers for Augmented Reality applications.

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

[1]Patent Application: INPI 20171000063377

II. Graphic Code

The Graphic Code coding system involves two main processes, the encoding and the decoding. Both receive as input one dictionary and grid specification (write and read order). Moreover, the encoding process also receives the message to be encoded, and when necessary, the base image. The decoding process, on the other hand, receives the image with the graphic code on it.

The main concept of the encoding process is to create an image with some kind information encoded on it by transforming structured information in pixel cells aided by a dictionary (conversion between symbols and pixel cells), and a grid specification (write and read order). The result is a marker that can subsequently printed and scanned.

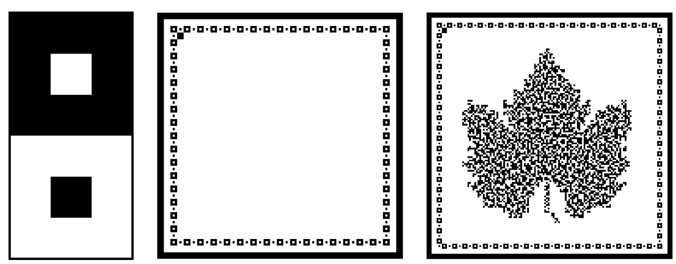

Fig. 1: Frame used to aid the decoding process of graphic codes from photos.

In order 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. 1, contains four nested elements: (1) a black rectangle with thickness of 3 pixels, (2) a white rectangle, also with thickness of 3 pixels, (3) 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 (4) a 3×3 black square at the top-left corner adjacent to the alternated squares patterns. Fig. 1 illustrates the alternating squares (left), a full frame (center), and a complete marker containing a coded image wrapped by the frame (right).

The detection process searches for this 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

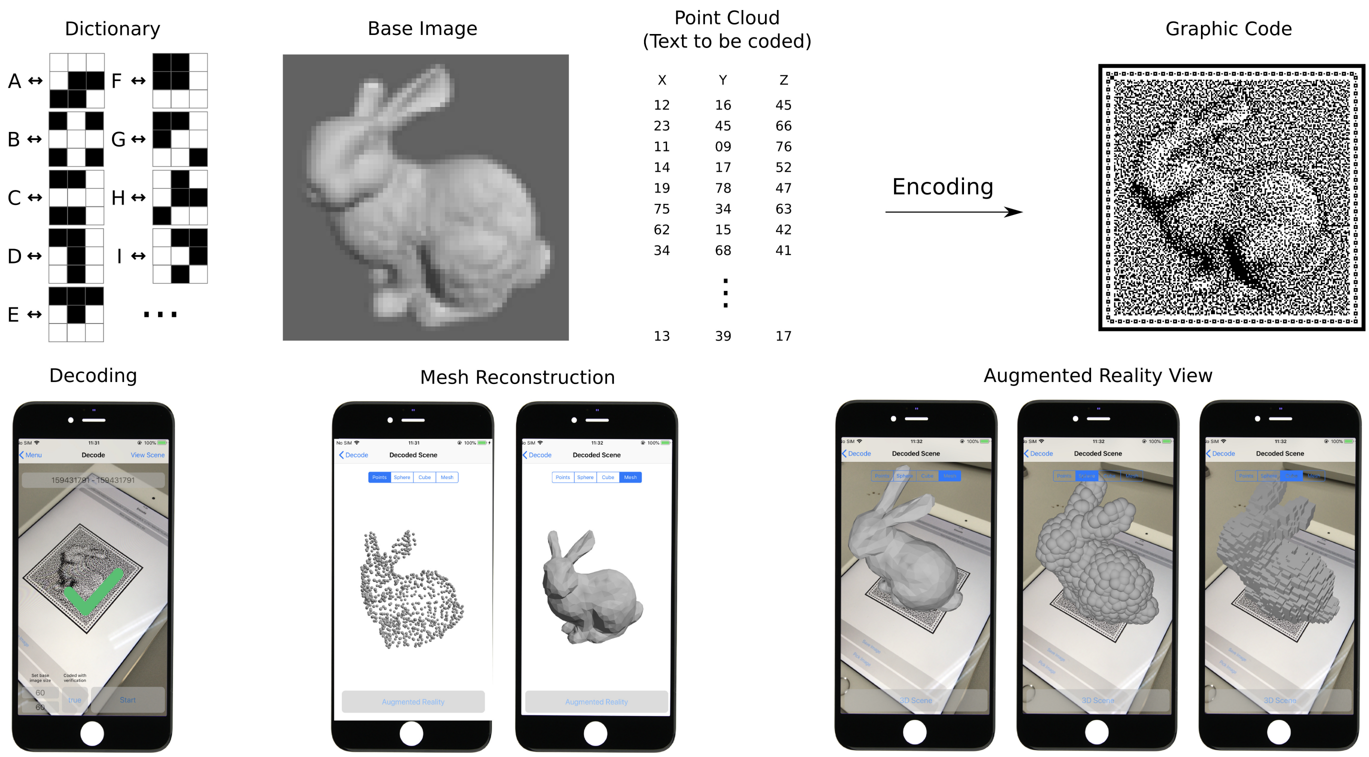

Fig. 2: Full pipeline to encode and decode information inside a marker for Augmented Reality applications.

the frame retrieved from the detection process. The purpose of rectification is to obtain an image in which the code appears without distortion and with such a known scale so that we calculate the translation and rotation between the marker plane and the respective camera plane. We also use the 3×3 black square at the top-left corner to define the marker orientation. Once we have the rectified marker image, the decoding process is the inverse of the previous explained encoding process. In this way, we go through the grid searching for patterns that are in the dictionary by employing the same encoding order. Whenever one of them is found, the respective symbol is concatenated to the message that is being retrieved. At this point and after the image rectification, we can also re-project a 3D model in the frame plane space using the inverse transformation matrix retrieved in rectification step. As a result, we finish our Augmented Reality application pipeline by overlapping this 3D model on the image. The full process of coding specifically to this Augmented Reality application is presented in Fig. 2.

III. Applications

In the example that is being presented, we are using 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 unstructured elements). Furthermore, all these models could be compressed allowing the coding of even larger models. The example presented in Fig. 2 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 the mesh from it).

IV. Conclusions

In this work, we have shown that Graphic Code can be more versatile than classical MRCs. The increased coding capacity of Graphic Codes allows the creation of applications that handle with larger models, like meshes, curves and images. As future work, we intend to code other types of data and explore the combination of 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. In addition, we would like to thank all the other members of the VIS Team for their helpful contributions.

References

[1] L. Cruz, B. Patro, and N. Gonalves, “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

[2] 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