Halftone Pattern: A New Steganographic Approach

Leandro Cruz [1], Bruno Patrão [1] and Nuno Gonçalves [1,2]

[1] Institute of Systems and Robotics - University of Coimbra, Portugal
[2] Portuguese Mint and Official Printing Office (INCM)

Figure 1: *Four examples of images containing messages coded using the proposed techniques.*

Abstract

In general, an image is worth a thousand words, but sometimes, words are the most efficient tool to communicate some information. Thereupon, in this work, we will present an approach for combining the visual appeal of images with the communication power of words. Our method is a steganographic technique to hide a textual information into an image. It is inspired by the use of dithering to create halftone images. It begins from a base image and creates the coded image by associating each base image pixel to a set of two-colors pixels (halftone) forming an appropriate pattern. The coded image is a machine readable information, with good aesthetic, secure and containing data redundancy and compression. Thus, it can be used in a variety of applications.

CCS Concepts

•***Computing methodologies*** → *Image processing;*

1. Introduction

In last decades, visual patterns able to code a machine readable information got huge importance due to the variety of applications. The first popular pattern was the 1D bar codes. Later, appeared the 2D barcodes, such as DataMatrix and QR Code. These patterns are able to code more information than 1D bar codes [Wav11]. However, none of them has a good looking (they look like a random displacement of black and white pixels). For appearance improvement, recently emerged some workarounds that change the coded image to create a model with a better overall appearance, but still able to decode the information according to the respective standard, such as Q Art Code [Cox12, CCLM13] (we will refer to these methods as *good-looking QR Codes*). In this work, we will present a new steganographic technique inspired on dithering, that has all good features of popular standards like QR Codes [Wav11], but is able to produce a better looking coded images.

Our approach† consists of creating a coded image by an appropriate arrangement of a set of two color pixels (halftone pattern), specifically placed in a particular order. This set of pixels is named cell, and the specific placement of pixels in a cell is named pattern. We choose some of those patterns and associate them to some symbols, that can be used to write the message to be encoded. This association is called dictionary. Our coding and decoding method are based on these elements.

The main contributions of this work are:

- Coding more information than good-looking QR Codes
- A coding and decoding technique able to create coded images with a good overall appearance
- Two other techniques (extensions of the first) able to create colored coded image

† Patent Application: INPI 20171000063377

DOI: 10.2312/egs.20181035

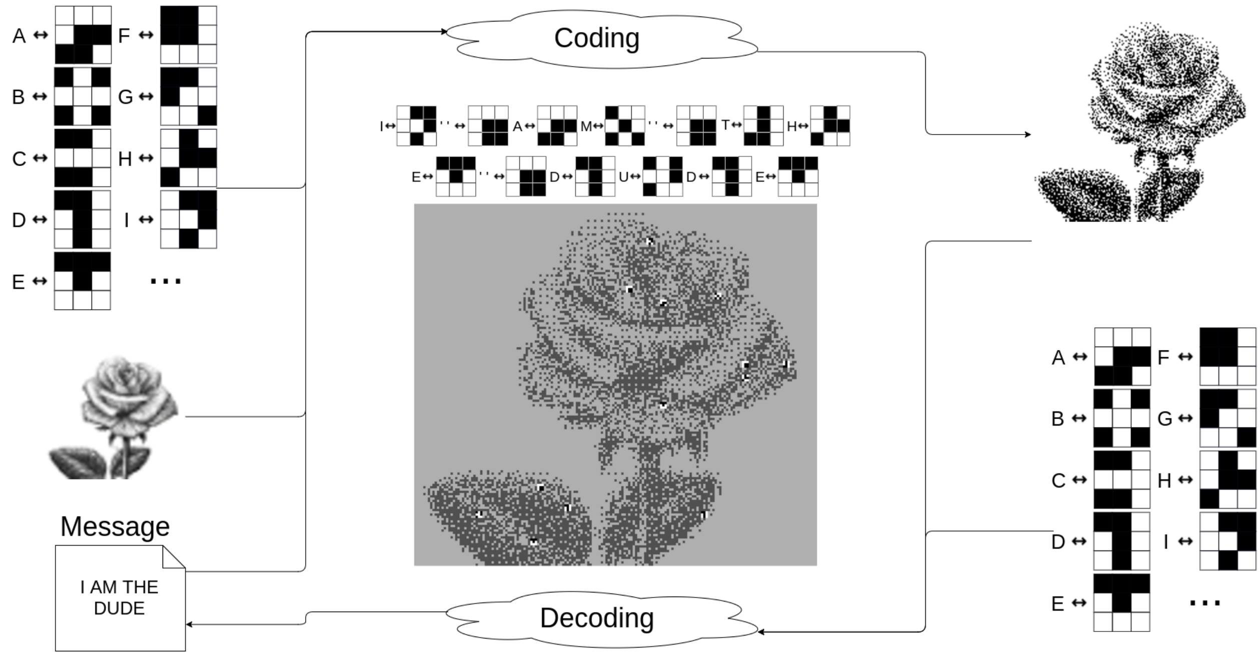


Figure 2: *Coding and Decoding Pipeline*

2. Related Work

Cryptography transforms the content of a message to hide the original content from all observers that doesn't have the necessary parameters, and does not allow them to find out the original content. Similarly, steganography is an approach for hiding an information (for instance, a text) in another model (for instance, an image). Our method fits in the second case.

Zhou et. al. [ZAC06] proposed a steganographic technique that hides a given image in several black and white images (halftone) that do not have visually recognizable features of the original (they are similar to a white noise). In our case, we use a base image to define the overall appearance of the coded image, and what our method codifies is a text. In addition, our decoding method recovers the exact coded information whereas, in the aforementioned technique, the decoded image contains some characteristics of the original image, but in general with additional noise.

A well-known steganographic approach was proposed by Rai et. al. [RGG07]. It performs a change of the Least Significant Bits (LSB) of some pixels for information encoding. Thus, it only creates small high frequency variations that produce imperceptible changes to human eyes. A limitation of this approach is the loss of information when the coded image is printed. It happens because the printing process also adds high frequency variations and it corrupts the coded information. Our approach diminishes the visual quality of the coded image in comparison to the base image because it reduces color space in a similar manner to dithering. However, it is robust to printing process and further recovering the image from a photo (under some reasonable hypothesis). Besides of the LSB approach, there are other steganographic approaches based on masks, filtering, and specific transformations, however a complete review of this subject is beyond the scope of this paper.

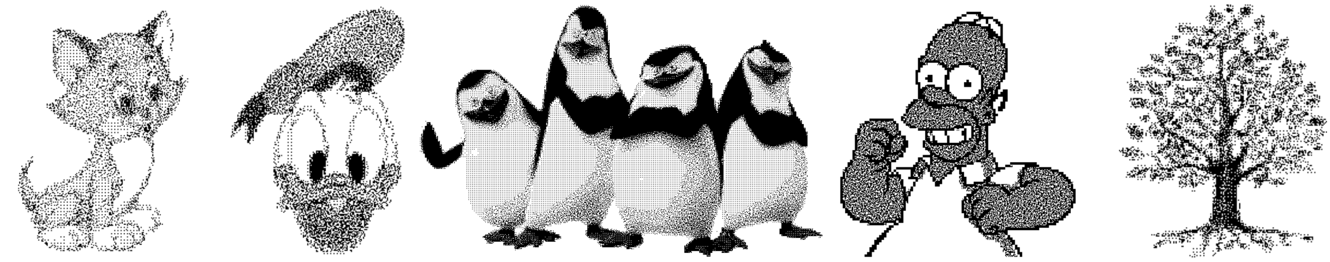
The basis of our approach is a binary coding: we choose a pair of colors (black and white, or any other). For recovering the coded image from a photo, it is important to choose a pair of high contrast colors. But there is not a strong constraint about the basic colors at first. Furthermore, we can extend this approach to use more than two colors in such way that we can define a binary pattern from a set of pixels. Figure 1 illustrates some of these possibilities.

Therefore, our method is able to produce coded images with a better aesthetic than that obtained by 2D barcode standards, like QR Code or Data Matrix. There are approaches that produce QR Codes models with more appealing elements, such as Q Art Code [Cox12] and Halftone QR Code [CCLM13]. From an aesthetic point of view, our approach produces results much better than those produced by such techniques.

3. Coding and Decoding

Dispersed dot dithering approach [Uli87] is the inspiration for our technique. It transforms each pixel of a grayscale image into $k \times k$ black or white pixels, and then, it reduces the color space (from 256 levels of gray to black or white) in exchange for increasing image resolution (multiplies each dimension by k), preserving human perception of grayscales. During dithering process a quantization of a grayscale image is performed. And each *quantum* (the quantized gray pixel) is transformed in a $k \times k$ black and white pattern.

Similar to dithering, our method is also based on an association of quantized color and pixel pattern. Each quantum is associated to many patterns, whose quantity distribution of pixel color is: in quantum Q0 all pixels are black; Q1: 1 black and $k^2 - 1$ whites; Q2: 2 blacks and $k^2 - 2$ whites; and so on. The basis of our process is a mapping between symbols and some of these patterns, named dictionary. The set of dictionary symbols defines an alphabet that can be used to create a message to be encoded and decoded. This alphabet can be binary, numeric, alphanumeric, unicode, etc.



(a) 52×76, 1822, 26ms and 24ms (b) 55×76, 1507, 29ms and 26ms (c) 78×52, 1321, 28ms and 24ms (d) 120×116, 4381, 88ms and 81ms (e) 60×75, 1500, 30ms and 28ms

Figure 3: *Five examples of coded images (using a drawing-type base image). For each example, we show: the quantity of base pixels (base image resolution), quantity of candidates, and the time to encode and decode (in milliseconds).*

Our coding and decoding approach are illustrated in Figure 2. The coding input is the dictionary, the message (only using symbols of the alphabet), and a base image. The first step is the identification of candidates: pixels in base image whose associated pattern has the quantity distribution of pixel color equals to the patterns of dicionary. Between them, we choose some (the quantity equals to the message size) to be used for encoding a message symbol. The coding method scans the cells in a specific order and when we find a chosen candidate, we insert in the coded image the pattern corresponding to the respective symbol. For unchosen cells, we introduce a pattern (according to its respective quantum) that does not appear in dictionary. Similarly, the decoding input is the coded image and the dictionary. The output will be the decoded message. It begins by scanning the image in the same order it was coded and, whenever we find a pattern contained in the dictionary, we concatenate the respective symbol on the message that is being retrieved.

This halftone approach can be straightforwardly extended to create colored coded images. Following, we will present two approaches for this purpose. In these cases, it is only important that the decoding method is still able to recognize the two basic elements from the colored pixels.

The first approach consists in to choose of one of the basic colors (for example: black) and vary its pixels in coded image while keep the other color's pixels. So, given a colored base image, we transform it in a grayscale image and perform the coding exactly as aforementioned. Then, for each cell, we change the color of black pixels to the respective color in colored base image. The Hommer Simpsosn picture, in Figure 1, was generated using this approach.

The second variation follows the same two-color per cell approach, but it is not dithering-based. The quantity of cells in coded image is the same quantity of pixels of a given colored base image. The distribution of pixel quantity does not matter, so we can define all cells as candidate. Then, we perform the coding (for each message symbol, to choose a cell and fill it according to dictionary; fill the rest of the image with patterns that are not in dictionary). Finally, for each cell, we change black pixels to a color a little bit darker than the respective pixel in colored base image, and white pixels to a color a little bit lighter. Figure 1 illustrates this approach using the Leonardo Da Vinci's Monalisa as colored base image.

The same image can contain more than one message (each one encoded using a different dictionary). In this case, the used dictionaries cannot have a shared pattern. Moreover, during the coding, we cannot change a cell filled in a previous step. The decoding process is the same: it just depends on the used dictionary.

The dictionary can contain one or more patterns for each symbol. In our tests, each symbol is associated to four 3×3 cells, each one containing, respectively, the following amount of black and white pixels: 6 / 3, 5 / 4, 4 / 5 and 3 / 6 (so, candidates are the pixels in base image whose respective pattern has one of these quantity distribution of pixel color). A $k \times k$ cell of black and white pixels (containing n whites) will have $C(k^2, n)$ (binomial of k^2 by n) different patterns. Because, $C(9,3) = C(9,6) = 84$ and $C(9,4) = C(9,5) = 126$ we can use a dictionary containing at most $83 = \min\{84, 126\} - 1$ symbols (-1 is related to the need of at least one pattern not used in dictionary that will be used to fill the unchosen cells). We use a dictionary with 41 symbols (26 letters, 10 digits, space, ".", ",", "!", "?").

To fill the remaining cells, we choose patterns outside the dictionary. We give priority to patterns that adapt to the shape of the base image in order to generate a better-looking image. For visual improvement, we can use a dithering approach [Uli87, VG91] in parts of image not filled by any pattern associated to the message. During this processing stage, it is necessary to guarantee that no pattern used in the dictionary will be introduced in the coded image.

4. Data Processing

One of the main QR Code features is the decoding robustness provided by data redundancy, using the Reed-Solomon algorithm [RS60]. We can adapt our method to include information redundancy, data compression, cryptography or any other procedure. For this purpose, we first create the cells using our dictionary correspondence (1st order cells). Then, we place all pixels in a binary vector and process it. We split the processed data in blocks of $k^2 - 1$ pixels, and we create cells whose first pixel is black and the others are the processed block (2nd order cells). Thus, we fill the coded image placing each 2nd order coded cell in a position according to the respective quantum (dithering-based) and message position. The other cells will be created according to respective base image quantum and will have a white pixel in the first position. The decoding method begins by identifying the cells with process data (with a black pixel in first position), vectorize this data, apply the inverse processing, re-create the cell with $k \times k$ pixels (according to the vectorizing order), and decode these cells using our decoding method. This approach is illustrated in Figure 4,

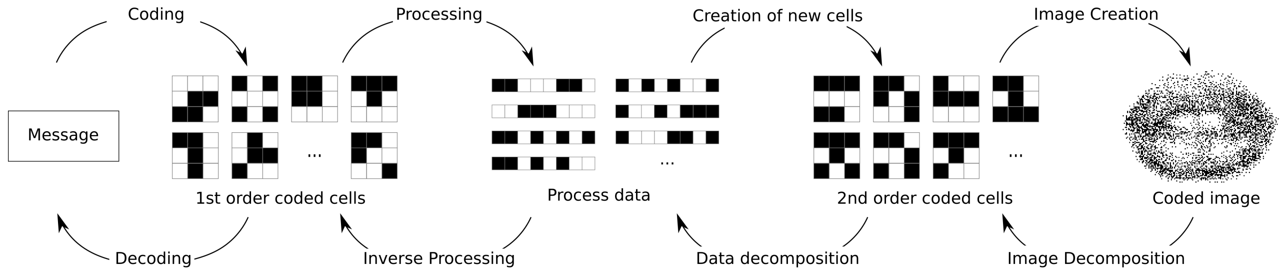


Figure 4: *Extended coding and decoding pipeline including data processing (like compression, redundancy, cryptography, etc).*

Another important QR Code feature is the capability to code considering amount of data. It codes at most 4296 alphanumeric characters, while DataMatrix can code 2355 [Wav11]. This limit is due to the limit of resolution specified in their standards. Using the same image size (177×177 pixels) and 3×3 cells, we can code 3481 characters (without compression). However, our method was designed to be better than the good-looking QR Codes. In these cases, the coded image has three times pixels in each dimension, and thus, to produce an image with 531×531 pixels we use 31329 cells (and then we can code at most 31329 symbols).

Whenever we choose for candidates according to base image's pixels quantum, we cannot use all cells as candidate. Figure 5 shows the average of the quantity of candidate pixels of five class of images (each one containing 60 images). Then, in average, we can code 15288 characters in an image with the same size of the larger good-looking QR Code (3,5 times more).

5. Conclusions

Although the proposed techniques are only based in two colors, there is no real constraint about this limit for coding and decoding. We opt for presenting only this case to keep focus on the relation between the base image colors and the pixels pattern. A future work is the creation of approaches to handle with more than two elements related to same hue or full colored base images. Even using a halftone pattern, we present two methods for creating colored coded images (an extension for 2D bar codes).

Our method, similarly to approaches such as QR Code and good-looking QR Codes, can be used in printed images and therefore it can be used for several applications. Most of steganographic approaches work only with uncompressed digital images. We kept focus on coding and decoding process and did not discuss about how to reconstruct the pixels from a photo of the printed coded image, because of the space limitations. But, it is worth emphasizing that in order to reconstruct the printed image we must use some suitable technique to detect the codified image, rectify it, to make an adequate sampling of each pixel and decide its color.

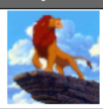
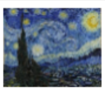
Icon	Drawing	Graphics	Painting	Natural
				
51%	40%	55%	53%	50%

Figure 5: *The average of the quantity of candidate pixels of five class of images.*

Of course, there are boundaries related to image reconstruction, such as: printing resolution (high resolution reduces the physical size of the pixel which makes reconstruction more difficult), printing quality (ink blurring and poor toner quality can affect the color recognition), acquisition quality (focus or lens distortions may affect the reconstruction). However, under some reasonable hypothesis, our approach has great potential in this direction. A proper study about this subject is another future work.

The coding and decoding time is low (Figure 3). Moreover, to obtain a beautiful coded image, in general, the base image is not very small, and then, we have several candidates (more than is usually need for this type of coded images). Consequently, we can add more redundancy, increasing the robustness related to image degradation: like compression, printing and photographic artifacts, dust, noise, etc (as well as it happens in QR Code models).

Despite of the information security not being our main purpose, the computational cost for the cryptanalysis of our method is very high, because of the high degree of freedom: which alphabet was used in dictionary, the pattern size (k), the writing order of cells along of image, the writing order of pixels along of cell, and the dictionary. Moreover, we can also use a more sophisticated cryptography approach in our coding process.

The presented technique introduces a new way to code information into images. Our approach has the advantages of well know steganographic methods, but improves the capability of keeping the aesthetic value of coded images and increases the amount of coded information. Then, it is a flexible approach that can be extended and used for several purposes.

References

[CCLM13] Chu H.-K., Chang C.-S., Lee R.-R., Mitra N. J.: Halftone qr codes. *ACM Transactions on Graphics* (2013). 1, 2

[Cox12] Cox R.: Q art code, 2012. URL: http://research.swtch.com/qart. 1, 2

[RGG07] Rai P., Gurung S., Ghose M. K.: Analysis of image steganography techniques: A survey. *International Journal of Computer Applications* (2007). 2

[RS60] Reed I., Solomon G.: Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics* (1960). 3

[Uli87] Ulichney R.: *Digital halftone*. MIT Press, 1987. 2, 3

[VG91] Velho L., Gomes J.: Digital halftoning with space filling curves. *SIGGRAPH* (1991). 3

[Wav11] Wave D.: Qr code essentials, 2011. 1, 4

[ZAC06] Zhou Z., Arce G. R., Crescenzo G. D.: Halftone visual cryptography. *IEEE Transactions on Image Processing* (2006). 2