# OPTIMIZATION OF THE VISUAL QUALITY OF INSERTED IMAGE IN A QR CODE BASED ON HALFTONE TECHNIQUES

By Ofelia P. Villarreal

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTY OF PHYSICAL-MECHANICAL ENGINEERING
SCHOOL OF ENGINEERING IN ELECTRICAL, ELECTRONIC AND TELECOMMUNICATIONS
BUCARAMANGA
2014

# OPTIMIZATION OF THE VISUAL QUALITY OF INSERTED IMAGE IN A QR CODE BASED ON HALFTONE TECHNIQUES

By Ofelia P. Villarreal

A thesis submitted to the School of Engineering in Electrical, Electronic and Telecommunications in partial fulfillment of the requirements for the degree of Master of Science in Electronic Engineering

Advisor:
Rodolfo Villamizar Mejía, Ph. D

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTY OF PHYSICAL-MECHANICAL ENGINEERING
SCHOOL OF ENGINEERING IN ELECTRICAL, ELECTRONIC AND TELECOMMUNICATIONS
BUCARAMANGA
2014

## ACKNOWLEDGMENTS

I want to specially thank my advisor Rodolfo Villamizar. He has helped me to achieve this important goal in my life. I am grateful to the Dean, the Faculty, and the Staff of the School of Electrical, Electronic and Telecommunications Engineering for providing their assistance and support during my master program.

I also want to thank my husband Henry for his support and encouragement during my studies. He provided me the strength and true love for every single day.

My family and friends, for their collaboration and company in this part of my life.

## Contents

## LIST OF FIGURES

# ABSTRACT


## TITLE

OPTIMIZATION OF THE VISUAL QUALITY OF INSERTED IMAGE IN A QR CODE BASED ON HALFTONE TECHNIQUES[1]

## AUTHOR

VILLARREAL Dulcey Ofelia Patricia[2]

## KEYWORDS

QR code, halftone, optimization

## CONTENT

This work introduces an automatic method to improve the visual quality of a QR code image by embedding images that can be visually recognized by users. The resultant QR code with a embedded image is compatible with standard decoding applications for smartphone available in the market. In order to obtain a visually improved QR code the luminance values of the original image to be inserted are manipulated to preserve the QR code information modules. By using halftone techniques a image with blue noise spectral characteristics is obtained and visual roughness due to the customization is decreased in the algorithm. Non-linear optimization techniques are used in order to optimize image luminance levels by sections. A error probability model was considered to guarantee a successful binarization during the decoding and also a Human Visual system was used to validate the quality of the embedded image. Finally a visual comparison between the proposed and existing methods is presented evidencing considerable visual quality improvement.


---

[1] Research work

[2] Faculty of Physical-Mechanical Engineering, School of Engineering in Electrical, Electronic and Telecommunications, Advisor:Rodolfo Villamizar Mejía

# RESUMEN


## TITULO

OPTIMIZACIÓN DE LA CALIDAD VISUAL DE UNA IMAGEN INSERTADA EN UN CÓDIGO QR BASADO EN TÉCNICAS DE MEDIOS-TONOS[3]

## AUTOR

VILLARREAL Dulcey Ofelia Patricia[4]

## PALABRAS CLAVES

QR code, medios-tonos, optimización

## CONTENIDO

En este trabajo se presenta un método automático para mejorar la calidad visual de una imagen de código QR mediante la incorporación de las imágenes que pueden ser reconocidas visualmente por los usuarios. La imagen embebida en el código QR es compatible con las aplicaciones de decodificación estándar para teléfonos inteligentes disponibles en el mercado. Con el fin de obtener un código QR mejorado visualmente los valores de luminancia de la imagen original a ser insertada son manipulados para preservar los módulos de información del código QR. Por medio del uso de técnicas de halftone se obtiene una imagen con características de ruido espectral azul y la aspereza visual debida a la personalización se reduce en el algoritmo. Técnicas de optimización no lineal son utilizadas con el fin de optimizar por secciones los niveles de luminancia de la imagen. Fue considerado un modelo de la probabilidad de error para garantizar una binarización exitosa durante la decodificación. También un Sistema Visual Humano fue usado para validar la calidad de las imágenes incrustadas. Finalmente, una comparación visual entre el algoritmo propuesto y métodos existentes es presentada evidenciando una considerable mejora en la calidad visual.


---

[3] Trabajo de Investigación

[4] Facultad de Ingenierías Físico-Mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Director: Rodolfo Villamizar Mejía

# Chapter 1

# INTRODUCTION

Barcode has emerged as an alternative for inventory tracking of products due to that access to digital information is easy and fast. Traditionally a barcode has a vertical lines arrangement, however in the last decade, it has evolved to a bidimensional version. This version has increased storage capacity and other features for including different types of information. There exist different bidimensional codes and one of the most representative is the Quick Response Code or QR code. This code was developed several years ago in order to do tracking and inventorying car parts, for that reason, it is characterized by achieving high decoding velocities and having high data storage capacity [8]. Those characteristics have allowed that QR codes represent an emergent communication strategy and have motivated several researches, specially for marketing problems, where it has become a powerful mobile's tool. For example, by means of a smartphone it is possible that an user can decoding information from a printed version scanned by using the phone's camera. Thus, the use of QR Code is exponentially spreading out in several areas, where different web applications are being developed, such as to read information from business cards, direct access to open web pages, videos download, to get discount by reading digital coupons, to access a virtual tour for house sales, to open text documents or to buy train tickets, among others [8]. A QR code consists of a bidimensional binary image what contains functions patterns and a encoding region, whose characteristics make it a valuable tool to involve technologically the user in digital services through printed formats. As shown in figure. 1.1 (a) the aesthetic appearance of the QR code is very rough, colour and form limited and does not offer any visual evidence about what information is into it. Thus, it is mandatory to scan it in order to know what type and who is the owner of information encoded. On other hand, if a special attention is desired from a QR code user, additional visual and coded information must be added. A possible approach is depicted in figure. 1.1 (b) where additional images and description are considered to create a previous

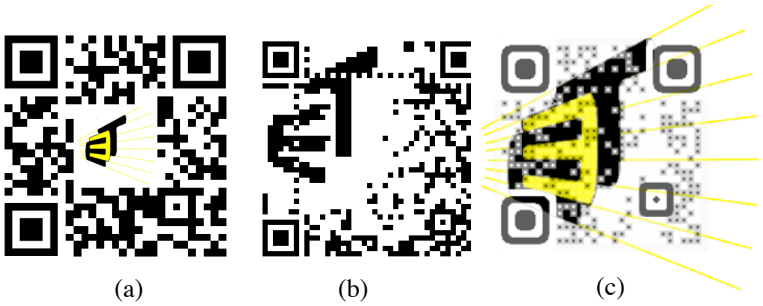

(a) (b)

**Figure 1.1:** QR code image. a) Original QR code image. b) QR code image used for advertising

visual information for the QR code user and it corresponds to the research problem considered in this work.

The QR code has a squared and monochromatic design that can be annoying when it is combined with a billboard or printed advertisement. Because QR design, to include visual information is a challenging task, by considering that different approaches have been developed about it. Most of approaches consist of replacing part of the encoding area by a non coded image (see figure. 1.2 (a)). Generally, this process is manually done, however in a recent application [9] automatic location of an image into a QR code encoding area is proposed but into a limited area, what produces a low decoding capacity and high noise sensibility. In [10, 1] tags and error correction capacity are used to relocate the modules in the encoding region. This technique encodes an image inside the QR code by relocating some QR modules as pixels of the image (see figure. 1.2 (b)). Another interesting approach is presented in [2] what consists of manipulating the image luminance values. In this approach, the QR code resultant contains a visually identifiable image but a coarse squared structure of modules persists with a binarized appearance, figure. 1.2 (c). Thus, this work proposes to embed automatically a image into a QR code, by using halftone techniques what allow to obtain a new coded image preserving similar appearance than the original one. The resultant QR code conserves the decoding robustness of a typical QR code, but with visual information and a improved graphical design.

Halftone is a technique that simulates a continuous tone image by binarizing each pixel and taking into account statistics information about the surrounding area where pixel



(a) (b) (c)

**Figure 1.2:** QR code approaches. (a) Permute modules in the QR code by an image [9]. (b) Alter the position of the modules by modifying tags in the encoded data [1]. (c) Modify the luminance of the image [2]

is binarized. This technique has been used for decades in printing industry and the main challenge is to determine the optimal distribution of pixels subjected to the blue noise criteria, described in the next chapter. The spectral characteristics of the resultant halftone images are an interesting alternative for improving the visual appearance of a QR code.

Thus, the principal contribution of this work is an approach to customize a QR code by embedding any image, by using halftone techniques subjected and including an error probability measurement.

## 1.1 Thesis Objectives and Organization

The objectives of this thesis are summarized in the following topics:

*Main objective*

- To customize a QR code image by integrating a chromatic image into it by using halftone techniques and preserving code integrity with minimal loss of encoding area

*Specific objectives*

- To integrate an image into a QR code, preserving code integrity and with minimal loss of encoding area
- To adapt halftone techniques to the QR coded image appearance improvement problem
- To determine the best halftone technique for improving the visual appearance of a image integrated into a QR code

In order to achieve these objectives, two conditions were taken into account: i) The resultant QR code must be decodable by commercial QR code readers. ii) Error correction capacity must be conserved.

This master thesis report is organized as follow: Chapter 2 describes the theoretical framework boarded in this thesis, where QR code's principles, its main structures, important features and decoding process are described. Also, halftone concept and techniques are presented. Chapter 3 presents the mathematical developments required to obtain the proposed algorithm. Chapter 4 explains the algorithm proposed and its main implementation results. Finally, Chapter 5 provides conclusions about the QR codes obtained with the proposed algorithm and suggests some future works to improve some drawbacks observed.

# Chapter 2

# BACKGROUND THEORY

## 2.1 QR code fundamentals

QR code is an improved bidimensional barcode with respect to another previously proposed such as: PDF147, which allows high data storage capacity, MAXICODE that supports high-speed decoding adn DATAMATRIX that keeps good data storage capacity and error correction characteristics. Thus, by using a QR code, an increased data storage capacity, diverse error correction levels, highest decoding speed and eastern characters supports can be obtained [11, 12, 13, 14]. The QR code's popularity has grown fast, especially in Asiatic countries, probably because eastern characters are supported. Particularly in Japan, the QR code regulation is developed under the Japanese standard JIS-XO510 (Japanese Industrial Standard-JIS, 1999). On a worldwide level, QR code became an International Standard by 2000 under the Norm ISO/IEC 18004:2000 [14]. Figures 2.1 (b), (c), (d) and (e) belong to PDF147, MAXICODE, QR code and Datamatrix symbols respectively.

QR codes were specially designed in the automotive industry for auto parts tracking, where high speeds and storage capacity is required. For that reason, the design of a QR

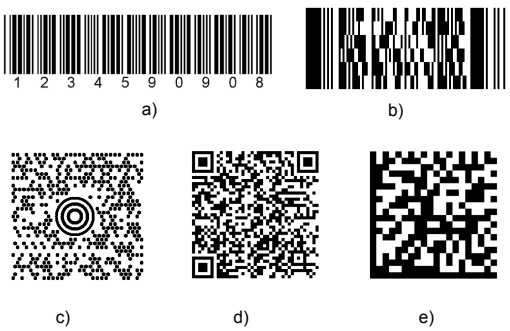



**Figure 2.1:** Barcodes symbols. a) Onedimentional barcode b) PDF147 2D-barcode, c) MAXICODE, d) QR code, e) Datamatrix. The symbols were created with the barcode generator tool developed in [3]

code symbol provides special patterns, mainly classified into two groups: function patterns and encoding region. Function patterns are used to improve detection speed and to correct misalignment in the QR code. Encoding region, on the other hand, is used for storing data and error correction information. Data in the QR code is represented by binary modules (squares in black and white), where black module represents a logical 1 and white modules represents a logical 0. Forty different QR code versions are found, where each one has associated an specific quantity of modules. Therefore, the amount of modules and version are directly proportional to the QR code data storage capacity. Figure 2.2 presents QR codes for versions 2, 6 and 12.


**Figure 2.2:** QR code version. (a) QR code version 2 of $25 \times 25$ modules, encoding E3T URL address. (b)QR code version 6 of $41 \times 41$ modules, encoding an e-mail message. (c) QR code version 12 of $65 \times 65$ modules, encoding a vCard contact information

Despite of plane appearance of QR codes, many smartphone applications have been developed, because they can be directly scanned. The detection process is depicted in figure 2.3. The first step consists of taking a picture of the QR code with a smartphone's camera, then the information is decoded by the QR reader software installed in the phone. Finally, the decoded information links directly to a webpage or special content in the Web.

**Figure 2.3:** QR code detection scheme

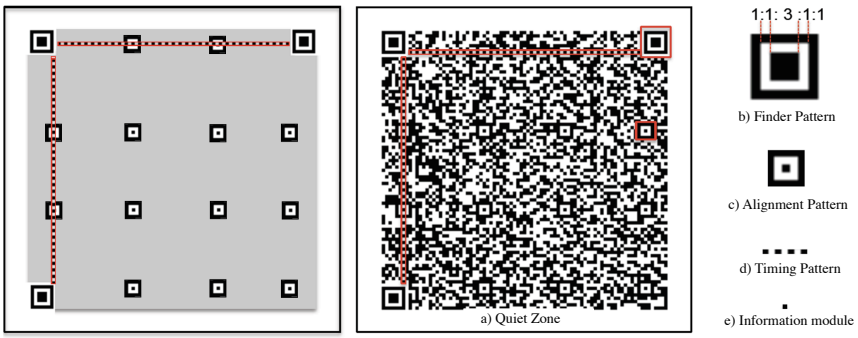



**Figure 2.4:** QR features scheme

### 2.1.1 Function Patterns

Function patterns are fixed structures defined inside the QR code and proposed as reference point to recognize quickly right positions of the QR code. They conform five different structures: **i) Finder patterns**. They are three concentric squares located in three of the four corners of the QR code. They keep the next modules ratio 1:1:3:1:1 and are used to locate an appropriated position of the QR code, even if it was read upside down (see figure. 2.4 (b)). **ii) Alignment patterns**. They are located along the encoding region and their quantity is related to QR code size and version. They are used in case of angles or distortions due to irregular surfaces and they help to relocate modules to their original position (see figure. 2.4 (c)). **iii) Timing patterns**. They are two lines of black and white modules (one vertical and one horizontal) and they are used to establish the module's center position in the symbol (see figure. 2.4 (d)). **iv) Format information**. It is located next to the finder patterns and they store information about version, size and mask used at the encoding stage.**v) Quiet zone**. It is the white zone surrounding the QR code and it separates the QR code from neighboring areas (see figure 2.4 (a)).

### 2.1.2 Encoding Region

It is the region chosen to storage information about data, version, format and error correction, what is shown in figure 2.4 in gray color. According to the QR code standard, the information to be encoded is transformed into a set of bits grouped by codewords. A codeword is represented by 8 bits or modules (white or black pattern). The data encoded by codewords

are organized sequentially into the encoding region. The sequential order is then altered by applying XOR operation between a previous defined mask and the encoding region, such as it is shown in figure. 2.5.

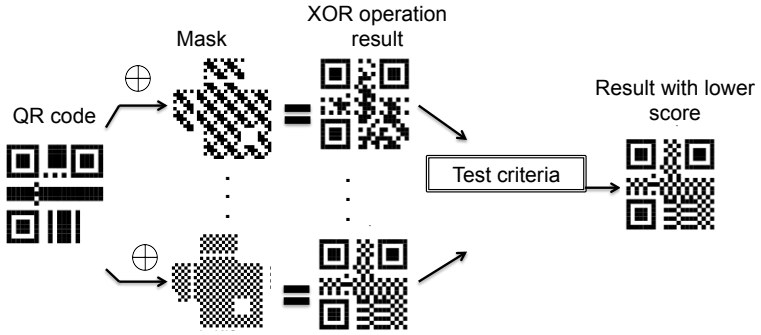



**Figure 2.5:** Masking process

The Masks are created by following conditions established in the table **??**. A module turns to black if such conditions are achieved. After XOR operation, the resultant matrix is evaluated and a score is assigned each time that a criteria is found. There are four criteria to be accomplished: i) Modules with the same color in rows/columns. ii) Block of modules with same color. iii) Alternate modules following the finder pattern ratio 1:1:3:1:1. iv) The proportion of black modules in the whole QR code area can't be more than 50 %.

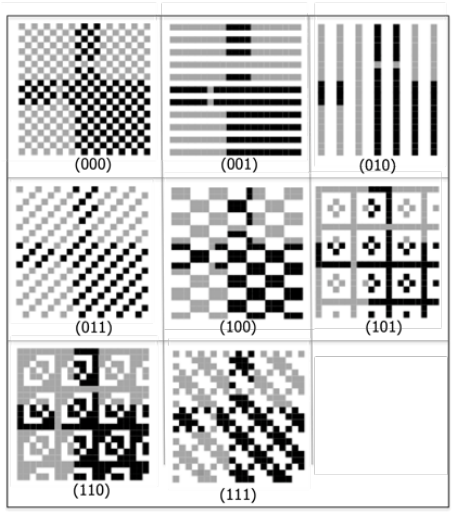



**Figure 2.6:** Mask defined by the QR Code Standard

Masking is necessary in order to avoid duplicity of combinations, that are exclusive for function patterns. This process is shown in figure. 2.5 and the masks defined in table 2.1 are shown in figure 2.6

| Reference | Conditions |
|---|---|
| 000 | $(i,j)mod2 = 0$ |
| 001 | $imod2 = 0$ |
| 010 | $jmod3 = 0$ |
| 100 | $((idiv2) + (idiv3)) = 0$ |
| 101 | $(i,j)mod2 + (i,j)mod3 = 0$ |
| 110 | $((i,j)mod2 + (i,j)mod3)mod2 = 0$ |
| 111 | $((i,j)mod3 + (i,j)mod2)mod3 = 0$ |

**Table 2.1:** Mask Generation Conditions

### 2.1.3 Error Correction and Storage Capacity

One of the advantages of a QR code is the high storage capacity and robustness to incompleteness of image. A QR code can store up to 7089 numeric characters, 4296 alphanumeric data, 2953 bytes data and 1817 kanji characters and they are available in forty versions, each one supports a specific number of modules. For example, first version's size is $21 \times 21$ modules while last version's size is $177 \times 177$ modules, a difference of four modules between consecutive versions is kept. Error correction capacity plays an important role to assure decoding of the symbol, what is made by a Reed-Solomon algorithm. A QR code image is captured under different environments, where distortion, damage and noise is expected. To avoid this, redundant information is included by means of error correction levels. Four error correction levels are known as L, M, Q, and H, where each one represents a percentage of 7%, 15%, %20, and 30% of the modules in a QR code, respectively. This percentage is known as the redundant information.

### 2.1.4 Decoding Procedure

Decoding process follows the same steps applied in encoding procedure, but two additional steps are added. i)Binarization. The image is acquired by the smartphone's camera in grayscale and then segmented in black and white pixels. This transformation is used to define the modules and extract the codewords. ii) Detecting the right orientation of the symbol and

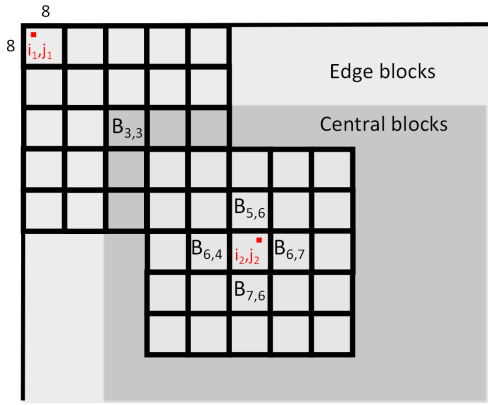



**Figure 2.7:** Block division used by Zxing library in thresholding calculation

the misaligned codewords are corrected. After these two steps, the decoding procedure follows the same steps as encoding but in backwards.

In order to achieve a precise binarization, depends on a good thresholding of the grayscale image. QR code standard does not specify any thresholding method in particular, thus different approaches are found in the literature about global and local methods. Global methods are based on the use of all pixels in the image in order to determine the threshold. However, under high noise or irregular illumination a poor performance is obtained [15, 16]. On other hand, local thresholding techniques consider a small predefined window of pixels. This method achieves better results under environments with irregular illumination [17, 18]. This work is based on the thresholding method used by the open source Zxing library, developed by google. This is one of the most popular libraries for QR code generation and decoding [19]. The thresholding method applied by Zxing library is based on the average luminance of pixels in a set of overlapping windows, combining global and local techniques. Thresholding method in Zxing library works as follows: The QR code image captured is divided in $8 \times 8$ pixels blocks, then luminance average is calculated for each block. Finally the blocks are grouped in windows of $5 \times 5$ blocks as shown in figure 2.7. The luminance average in the window is calculated as follows by equation 2.1,

$$\mu_{m,n} = \frac{1}{25 \times 64} \sum_{p=m-2}^{p=m+2} \sum_{q=n-2}^{q=n+2} \sum_{(k,l) \in B_{p,q}} Y[k,l]. \tag{2.1}$$

The average is calculated for each block $B_{m,n}$ and it is assigned to each pixel in the block as $t_{i,j} = \mu_{i,j} \in B_{m,n}$. The binary image is obtained by thresholding the luminance value at location $(i, j)$ of the image by using equation 2.2

$$Q_B[i,j] = \begin{cases} 1 & \text{if} \quad Y_{i,j} > t_{i,j} \\ 0 & \text{if} \quad Y_{i,j} \leq t_{i,j} \end{cases} \tag{2.2}$$

This thresholding technique is used to calculate the error probability and the procedure is explained in chapter three.

## 2.2 QR Code Customization Techniques

### 2.2.1 Direct Image Embedding

Direct image embedding is one of the first techniques used to QR code visual customization (see figure 2.8). It is described as follows: The redundant information area is replaced by images or logos, however, to replace by means of images can affect the QR code readability. In [20] an algorithm based on real-coded genetic algorithms is proposed to find the right position, scale and angle of the image to be inserted, without affecting the QR code readability. It is depicted in figure 2.8.

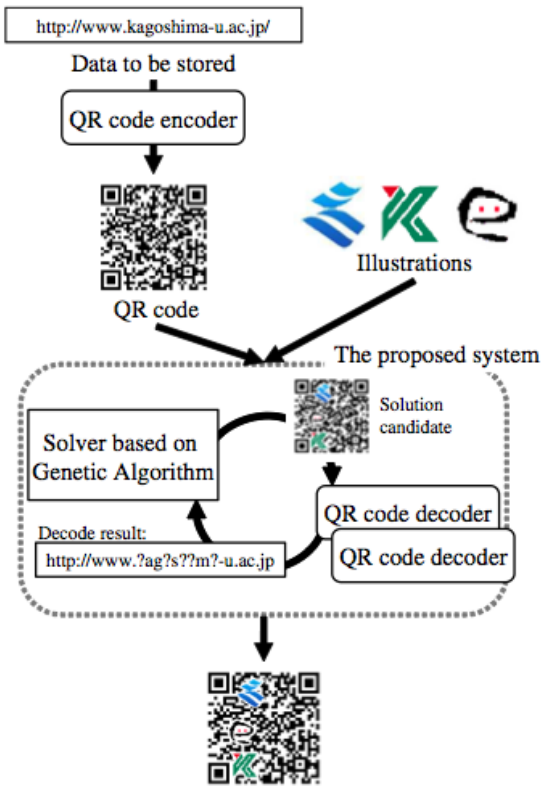



**Figure 2.8:** Genetic algorithm used to insert images into QR codes

### 2.2.2 Image Multiplexing

In the algorithm proposed by [4, 21], a dotted version of an image is inserted into the QR code, by taking advantage of the redundant information and padding codewords. The information is relocated in QR encode region, following the next steps: Initially, the QR code framework is drawn including function patterns, alignment patterns, quiet zone, version and error correction level (see figure. 2.9 (a)). Then the codewords and error correction words are added to the QR code, as depicted in figure 2.9(b). Traditionally the free area should be filled up with padding codewords and masking is applied, however in this step, the dotted version of the image is drawn in this free area (see figure 2.9 (c)). The size of the free space area is directly related to the error correction level chosen (see figure 2.10).

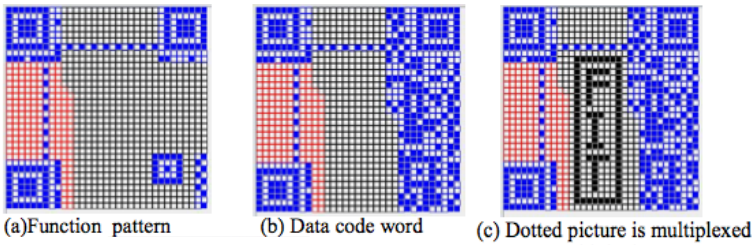



**Figure 2.9:** Multiplexing process [4]

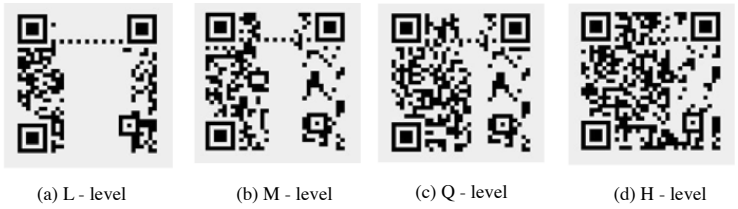



**Figure 2.10:** Error correction codewords distribution in a QR code [4]

This method presents the following drawbacks: i) Storage capacity has to be reduced to assure enough free space in the decoding region. ii) The error correction capability is decreased and consequently detection and decoding capacity is reduced, in comparison with the original version.

### 2.2.3 Reed-Solomon Modification

A non-systematic encoding of Reed Solomon Code to manipulate the parity symbols that are originally fixed, is presented in [5]. Reed Solomon code is a Maximum Distance Separable code (MDS), thus it is satisfied that a codeword non-systematically coded is equivalent to one systematically coded. Thus position of information blocks at the QR code is not restricted. The Reed Solomon (RS) codewords with smallest hamming distance among the modules are selected and placed on the QR code in an interleaved order. A predetermined masking pattern is developed to include the information symbols (see figure 2.11).

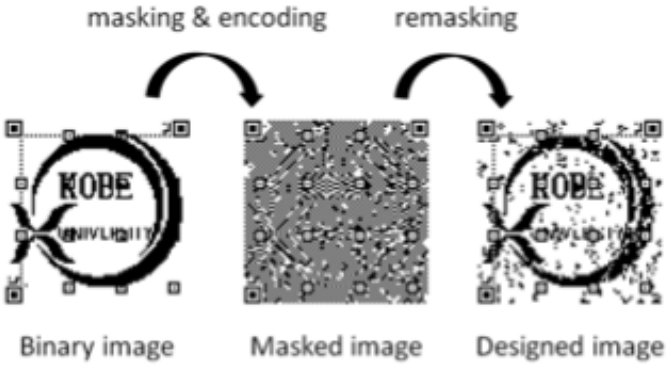



**Figure 2.11:** Reed-Solomon manipulation to increase the area for image customization [5]

## 2.3 Halftone Techniques

A halftone is a distributed set of dots that can represent an image that preserves fine details visible by the human (see figure 2.12). To achieve an optimal distribution of dots is not easy, because many displays and printing devices are unable to reproduce the distribution of those isolated dots without introducing printing artifacts (moire patterns) that decrease the quality of the image details. An example of those image artifacts is depicted in figure 2.13. The visual impact is minimized when the patterns concentrate most of the energy in higher frequencies where human visual system is less sensible. Different algorithms have been created to generate distributions with particular spectral properties. For example, Blue-noise mask generates high frequency binary patterns by thresholding a carefully designed multilevel array. The resultant pattern is aperiodic, radially symmetric (isotropic) and contains high frequency spectral components.

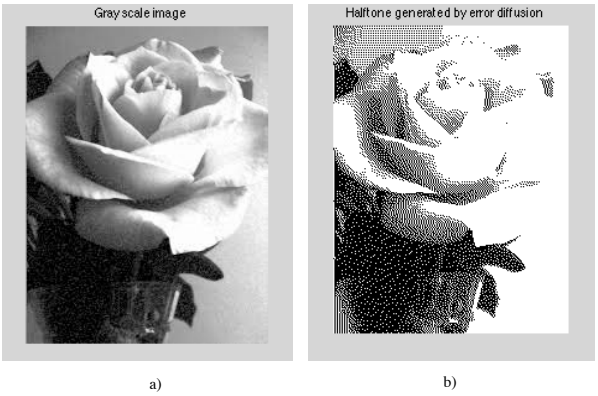



**Figure 2.12:** Halftone technique applied to a grayscale image

In this work, halftone techniques are used to select the pixels of the image that need to be modified according to the information content in the corresponding QR code module. Consequently, the squared appearance of the plotted image into the QR code, is minimized. Halftone technique takes advantage of the Human Visual System (HVS) optical characteristics, because lowpass nature of the spatial sensitivity. The eye blurs dots spatially close, thus a pattern discontinuos is perceived as a continuous tone and a pleasant visualization is obtained [22]. For blue-noise patterns the pixels are separated by an average distance $\lambda_b$, given by equation 2.3,

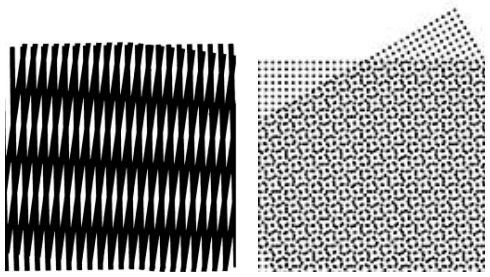


**Figure 2.13:** Artifacts produced in printing

$$\lambda_b = \begin{cases} \frac{D}{\sqrt{g}} & for \quad 0 < g \leq \frac{1}{2} \\ \frac{D}{\sqrt{1-g}} & for \quad \frac{1}{2} < g \leq 1 \end{cases} \tag{2.3}$$

where, $g$ is the gray tone level and $D$ is the minimum distance among points on the display. Among the Halftone algorithms proposed in literature to create blue-noise halftoning masks are: void-and-cluster, direct binary search and techniques based on centroidal voronoi tessellations [23, 24, 25]. Another blue-noise generator approach is the error diffusion developed by Ulichney [26], where excellent results are obtained, however because it is a recursive method, high computational resources are required.

### 2.3.1 Error diffusion

This approach was developed by [26] in 1989. In this technique the error accumulated in past iterations (quantization error) is distributed to non processed neighbor pixels. This algorithm is depicted in figure 2.14 and described by:

$$v[n] = \begin{cases} 1, & if \qquad (w[n] + w^e[n]) \geq 0 \\ 0, & else, \end{cases} \tag{2.4}$$

where,

$$w^e[n] = \sum Mi = 1 c_i \cdot v^e[n-i] \tag{2.5}$$

where $c$ is the error filter defined by Floyd's and Steinberg's [6] as,

| | | |
|---|---|---|
| | · | 7/16 |
| 3/16 | 5/16 | 1/16 |

Other error filters have been proposed with not better results than achieved by error diffusion. However, because this is a recursive algorithm it is not convenient when computational resources are low or in process requiring speed. One of the advantages of this technique is that error diffusion tends to enhance edges in an image (i.e. images with text can be better visualized).

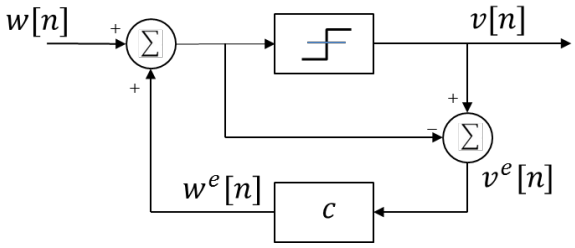



**Figure 2.14:** Error diffusion algorithm in [6]

### 2.3.2 Void-and-Cluster (VoC) Dithering Arrays

Void-and-cluster refers to the arrangement of minority pixels on the background of majority pixels. Void refers to a large physical space between minority pixels and cluster refers to close grouping of majority pixels. To obtain an homogeneous distribution of $0$s and $1$s implies add or remove minority pixels on the largest voids and clusters. Void and cluster focus in the ranking of pixels' location in the array. It is necessary to use two matrices in order to improve the ranking process. First, the Dither array is a matrix of $M \times N$, where the value assigned for each element, is a location where a 1 value will be assigned to the image. The Prototype Binary Pattern (PBP) matrix [7] has the same size as Dithering matrix, and helps to the ranking process, what includes the values between $0$ and $(MN - 1)$. By taking into account that minority of pixels corresponds to less than half of the pixels in PBP valued $1$ and majority of pixels are the remaining $0$s pixel, Ulichney in [7] proposed to use a filter in order to find the void-and-cluster groups of pixels. An special characteristic of the filter is that it must consider the neighborhood of each pixel in the PBP. Void-and-cluster scheme is described in [7] by the following equation 2.9,

$$D(x,y) = \sum_{r=-M/2}^{M/2} \sum_{s=-M/2}^{M/2} PBP(r',s')f(r,s) \tag{2.6}$$

where,

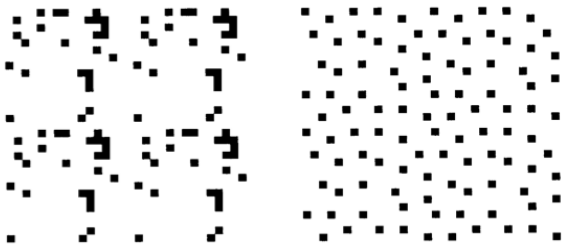
$$r' = (M + x - r)\textbf{module}M \tag{2.7}$$

$$s' = (M + x - s)\textbf{module}M \tag{2.8}$$

and $D(x, y)$ corresponds to Dither array, $PBP(x, y)$ is the Prototype Binary Pattern and f(r,s) is a two-dimensional Gaussian filter defined by the equation

$$e^{-\frac{x^2}{2\sigma_x^2}-\frac{y^2}{2\sigma^2 y}} \tag{2.9}$$

The arrangement results in $16 \times 16$ input pattern is depicted, in figure 2.15



a) Input pattern b) Resultant Rearrange pattern

**Figure 2.15:** Arrangement result of the void-and-cluster method using a two-dimensional Gaussian filter for a $16 \times 16$ input pattern obtained in [7]

In figure 2.16 (a), (b) a comparison between void-and-cluster and error diffusion method is shown. Void-and-cluster method obtain good results in halftoned images, however, due to the sharpening properties of error-diffusion, better results are obtained with this method in comparison with void-and-cluster.

a) Error diffusion method b) Void-and-cluster method

**Figure 2.16:** a). Void-and-cluster method for halftoning lena image b) Error-diffusion for halftoning the lena image made in [7]

### 2.3.3 Direct Binary Search (DBS)

This technique defines optimal patterns by incorporating a human visual system (HVS) model, which decides the dot to be kept in the halftone image, in order to represent the original continuous-tone in the most accurate way.

The process starts with a basic halftone image, where each pixel is processed. Then, by taking into account a $3 \times 3$ window, each pixel can be swapped by one of the pixels in the window or commuted from 1 to 0 or vice versa. An example of this process is presented in figure 2.17. The human visual system (HSV) $h(x)$ is defined as a continuous space linear filter whose output is defined in [27] as

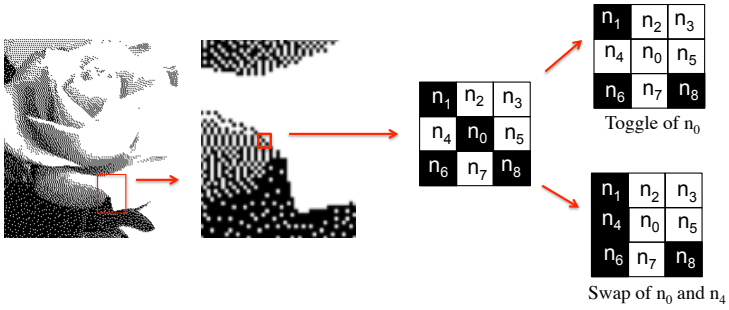



**Figure 2.17:** DBS algorithm for processing each pixel

$$\tilde{z}(x) = z(x) * h(x) = \sum_{n} g[n]\tilde{p}(x - Xn) \tag{2.10}$$

where,

$$z(x) = \sum_{n} z[n]p(x - Xn) \tag{2.11}$$

and

$$\tilde{p}(x) = p(x) * h(x), \tag{2.12}$$

where $g(x)$ is the halftone image and $p(x)$ is the distortion introduced in the printing process of the halftone image. The error between $z(x)$ and the original image $f(x)$ printed is

calculated as

$$f(x) = \sum_n f[n]p(x - Xn) \tag{2.13}$$

and

$$\tilde{f}(x) = \sum_n f[n]\tilde{p}(x - Xn). \tag{2.14}$$

The error is given then by

$$\tilde{e}(x) = \tilde{f}(x) - \tilde{z}(x). \tag{2.15}$$

By swapping or toggling pixels in the $3 \times 3$ window, the overall error measured in equation 2.15 is taken into account and this iterative process ends when error changes are no significant. However, this process requires significant computational resources, because for each pixel swappping or toggling the halftone image error must be recalculated.

### 2.3.4 Centroidal Voronoi Tessellated (CVT) Halftone Masks

In a good binary pattern, according to [28], minority of pixels has an aperiodic and isotropic distribution and it does not contain low frequency spectral components. Also, the minimum distance between a minority pixel and their neighbors must be computed by means of equation 2.16.

$$\lambda_p = \begin{cases} \frac{1}{\sqrt{g'}}, & for \quad 0 < g' \leq \frac{1}{4} \\ 2, & for \quad \frac{1}{4} < g' \leq \frac{3}{4} \\ \frac{1}{\sqrt{1-g'}}, & for \quad \frac{3}{4} < g' \leq 1 \end{cases} \tag{2.16}$$

In order to generate the mask, this technique performs a global optimization of the minority pixels by using CVT at each mask generation step. Particularly, Lloyd algorithm (It generates Voronoi Tessellation patterns) is used to locate a point that satisfies the $\lambda$ constrain defined in equation 2.16. In order to recover one particular pattern by means of threshold, a set of stacked binary patterns must be created . Furthermore, in order to accurately represent a given gray level, all the set must have a minimal concentration of minority pixels defined by $E[I_g[n] = g/G - 1]$. Finally a joint optimization of the set of binary patterns is performed to

assure the quality of the whole set. The halftone mask obtained (see figure 2.18) is defined by equation 2.17.

$$S = (\frac{G-2}{2})1_{M\times N} + B - W \tag{2.17}$$

where B and W are defined as:

$$B = \sum_{g=1}^{(G-2)/2} I_g^b \tag{2.18}$$

$$W = \sum_{g=1}^{(G-2)/2} I_g^w \tag{2.19}$$

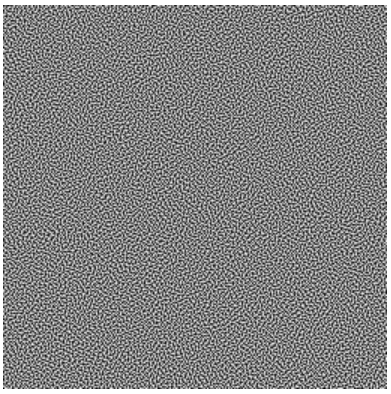


**Figure 2.18:** Halftone mask obtained by using voronoi tessellation

The computational complexity of this technique is between those of VoC and DBS. Furthermore, the resultant mask achieves better results than those obtained with VoC or DBS. Thus, in this work the CVT approach is used, because the resultant image has a better quality, than the obtained with other approaches. On other hand, the computational complexity of halftone pattern generation does not affect the speed of the QR code optimization process, because the mask design process is performed offline. In figure 2.18 a mask obtained with CVT is depicted.

# Chapter 3

# QR CODE VISUAL OPTIMIZATION PROCESS

This chapter describes the mathematical concepts and techniques used in the proposed algorithm, whose main steps are presented in figure 3.1. This approach consists of embedding an image into all encoding area of a QR code, where modifications of the smartphone' scanning software are avoided and most of applications available in the market are allowed. The main trade-off presents in this approach is between decodability and image visual quality, because the first one is affected by changing luminance values in each pixel of the QR coding area.

## 3.1 Pixel selection

A successful decoding can be assured if a whole information of the module is available during the scanning. However, the information contained in the central pixel of the module could be enough during the decoding process. Thus, this work considers only a few of pixels around the center of the module containing the information and the remainder pixels in the module follow the halftone mask distribution, restricted to the concentration of pixels modified $C_p$. The binary pattern generated by the halftoning mask will be denoted by $M_{C_p}$ and the

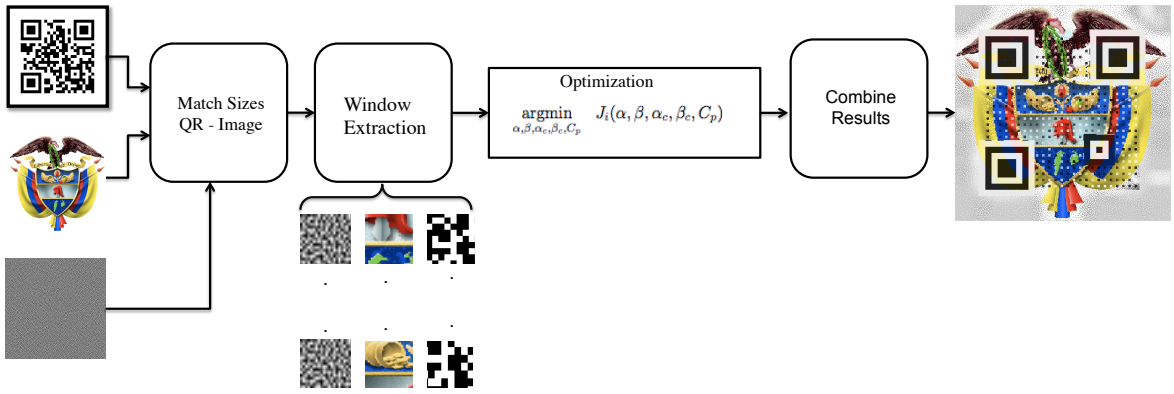



**Figure 3.1:** Customization process

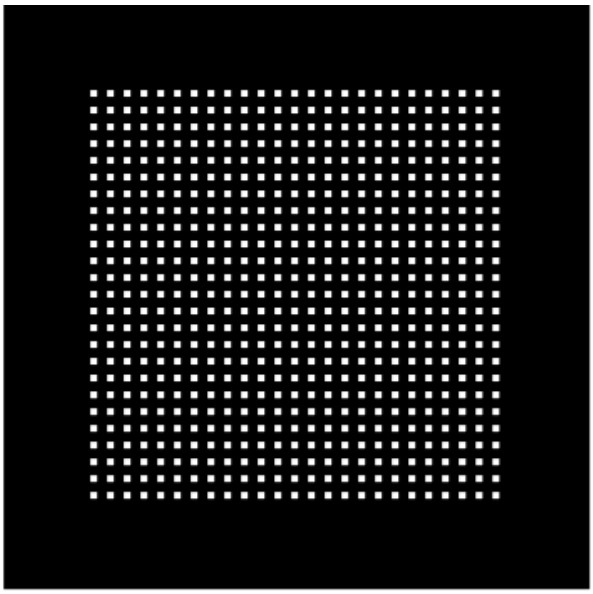


**Figure 3.2:** Central Pixels defined in $M_c$

central pixels of each module are previously defined with a Mask called $M_c$, which is 1 on the set of central pixels and 0 otherwise. An example of this is depicted in figure 3.2.

## 3.2 Luminance Modification

The luminance values of the pixels must reach one of the four possible optimum levels $\beta$, $\alpha$, $\beta_c$ and $\alpha_c$. Those levels are calculated for each patch in the image (They are divisions of image during the local optimization process, following equation 2.1) and its optimal values are obtained through the optimization algorithm described later in this chapter. The luminance of the embedded image $Y_{i,j}^{out}$ is selected based on the QR code module and the luminance values of the original image $Y_{i,j}$ at the point $(i, j)$ as described by equation 3.1.

$$Y_{i,j}^{out} = \begin{cases} \beta & \text{if } M_{c_{(i,j)}=0}, \quad q_{i,j} = 1, \quad M_{c_p(i,j)} = 1 \\ \alpha & \text{if } M_{c_{(i,j)}=0}, \quad q_{i,j} = 0, \quad M_{c_p(i,j)} = 1 \\ \beta_c & \text{if } M_{c_{(i,j)}=1}, \quad q_{i,j} = 1, \\ \alpha_c & \text{if } M_{c_{(i,j)}=1}, \quad q_{i,j} = 0, \\ Y_{i,j} & \text{otherwise.} \end{cases} \tag{3.1}$$

### 3.2.1 Color Optimization

The images to be embedded are originally in the $RGB$ color space, however if luminance values are changed by maintaining color characteristics it is necessary to use a different color space. First, it is necessary to measure the color differences into a perceptually uniform color space, what helps to fulfill the luminance constraints. For that reason, in this approach the luminance values are modified in the $HSL$ color space. This color space is usually used in computer graphics, because its transformation is not computationally complex. Other color spaces like $Lab$ color space could be used, however iterative calculations are necessary during the optimization process. In the $HSL$ color space the luminance value is manipulated up to obtaining a desired value, while the saturation and hue values are kept. This work use the luminance definition given by equation 3.2

$$Y = 0.2989R + 0.5870G + 0.114B \tag{3.2}$$

where the relation between equation 3.3 and the $L$ component in the $HSL$ color space is a linear and monotone function defined as $Y = f(L)$. The weight luminance vector can be defined like $v = [0.298, 0.5870, 0114]$ and $f(L)$ can be defined as follows

$$Y = f(L) = v^T T^{-1}(H, S, L). \tag{3.3}$$

The optimal value of $L$ is obtained through the luminance target $Y^{out}$ as the solution of equation 3.4

$$L^* = \underset{L}{\text{argmin}} |f(L) - Y_{out}|. \tag{3.4}$$

Once the optimal value $L^*$ is achieved, the pixel color components in $RGB$ color space are computed by using the forward transformation defined by equation 3.5

$$(R^*, G^*, B^*) = T(H, S, L^*). \tag{3.5}$$

### 3.2.2 Error Probability

The new QR code created by means of the proposed algorithm [29], compared with the typical monochromatic QR version, has a greater error probability during the decoding process because binarization depends on luminance threshold selected. Efficient optimization methods require that the computation of the error probability, defined in local neighborhoods in order to attenuate the parallel nature of the problem, be simplified. Thus, the probability model used can be integrated into different optimization algorithms, where two cases are treated independently: the first correspond to the probability of sampling incorrectly any value at the QR code module and the second corresponds to the probability of detecting the centers of QR code modules.

#### *A. Binarization Error Probability*

It is the probability of generating an incorrect binary value during the sampling process at any pixel in the QR code module. This probability is affected by diverse factors such as the local distribution of luminance values in the image, the distribution of pixels in the QR code and the luminance changes. In order to define the error probability, lets chose a pixel in the location (i, j) to detect its binary value. The probability of assigning a wrong binary value is defined by the total probability described in equation 3.6,

$$P_m = P(\text{define } Q_{i,j} = 1|Q_{i,j} = 0)P(Q_{i,j} = 0) + P(\text{define } Q_{i,j} = 0|Q_{i,j} = 1)P(Q_{i,j} = 1) \tag{3.6}$$

where $Q_{i,j}$ represents the pixel (i,j) of the QR code module. This pixel is read by a QR scanner based on the threshold value, given by $\mu_{m,n}$ and calculated in (2.1), and the luminance value $Y$. By applying it to the conditional probability, it is rewritten in terms of luminance and threshold as shown in (3.7),

$$\begin{aligned} P(\text{select } Q_{i,j} = 1|Q_{i,j} = 0) &= P(Y_{i,j}^{out} > \mu_{i.j}|Q_{i,j} = 0) \\ P(\text{select } Q_{i,j} = 0|Q_{i,j} = 1) &= P(Y_{i,j}^{out} < \mu_{i.j}|Q_{i,j} = 1). \end{aligned} \tag{3.7}$$

The exact computation of probabilities in (3.7) involves the knowledge of the joint probability distribution of the image and the threshold values. Since, these last in practice can be quite difficult to be calculated, a simplified model is proposed in this work by assuming that components of luminance $Y_{i,j}^{out}$ for the embedded image, are independent. Thus, the value of $Y_{i,j}^{out}$ at location $(i,j)$ can be decomposed as the combination of several random variables related to the luminance parameters of the transformation. Thus, each pixel is considered as non-central pixel and it is assumed that only two independent values $\alpha$ or $\beta$ are present.

The binarization threshold at location $(i,j)$ is in general given by (2.1) and it can be divided into three components $t_{i,j} = a_{i,j} + b_{i,j} + \eta_{i,j}$ with

$$\begin{aligned} a_{i,j} = & \frac{1}{25\times 64}\sum_{p,q}\sum_{\substack{(k,l)\in B_{p,q}\\(k,l)\in M_{cp}}} \beta Q_{k,l} + \alpha(1-Q_{k,l}) \\ b_{i,j} = & \frac{1}{25\times 64}\sum_{p,q}\sum_{\substack{(k,l)\in B_{p,q}\\(k,l)\notin M_{cp}}} Y[k,l], \end{aligned} \tag{3.8}$$

where $a_{i,j}$ corresponds to the pixels to be modified, $b_{i,j}$ the remaining pixels and $\eta_{i,j}$ is the noise. The component $a_{i,j}$ can be modeled as a binomial distribution assuming independence of the code pixels given by

$$P(a_{i,j} = x) = \sum_{k=0}^{n} \delta\left(x - \left[\frac{k\beta}{25\times 64} + \frac{(n-k)\alpha}{25\times 64}\right]\right)\binom{n}{k} p_1^k p_0^{n-k} \tag{3.9}$$

where $p_0$ is an abbreviation of $P(Q_{i,j}=0)$, $p_1$ correspond to $P(Q_{i,j}=1)$, $k$ is the number of pixels with modified luminance $\beta$ in the local window and $n$ is the total number of modified pixels. Since the variables $Y_{i,j}$ are assumed to be independent and identically distributed, the distribution of $b_{i,j}$ can be approximated by a Gaussian distribution denoted by $P(b_{i,j} = y) = f_Y(y)$ with mean $\mu_y = \frac{E[Y_{k,l}](25\times 64 - n)}{25\times 64} = E[Y_{k,l}](1-C_p)$ and variance $\sigma_y^2 = Var[Y_{k,l}]\frac{1-C_p}{25\times 64}$. By combining these expressions and using the fact that $a_{i,j}, b_{i,j}$ and $\eta_{i,j}$ are independent, the probability distribution of the threshold $t_{i,j}$ is given by

$$\begin{aligned} P(t_{i,j} = t) &= \sum_{k=0}^{n} (f_Y * f_n)(t - t_k)\binom{n}{k} p_1^k p_0^{n-k} \\ &= \sum_{k=0}^{n} f(t - t_k)\binom{n}{k} p_1^k p_0^{n-k} \end{aligned} \tag{3.10}$$

where $t_k = \left[\frac{k\beta}{25\times 64} + \frac{(n-k)\alpha}{25\times 64}\right]$ are the possible outcomes of the sum of $n$ modified pixels. The

distribution $f_\eta(n)$ corresponds to the Gaussian noise and $f = f_Y * f_n$ is the convolution of the image average values and noise distributions. Since $Y$ and $n$ are independent and Gaussian, $f$ is also Gaussian with mean $\mu = \mu_Y$ and variance $\sigma = \sqrt{\sigma_Y^2 + \sigma_n^2}$.

Finally, the conditional probabilities on (2.1) are obtained by conditioning on the set of modified pixels $M_{C_p}$ as follows:

$$\begin{aligned} P(Y_{i,j}^{out} > \mu_{i.j} | Q_{i,j} = 0) = P(\alpha > \mu_{i.j}) C_p \\ + P(Y_{i,j} > \mu_{i.j})(1 - C_p) \end{aligned} \tag{3.11}$$

and

$$\begin{aligned} P(Y_{i,j}^{out} < \mu_{i.j} | Q_{i,j} = 1) = P(\beta < \mu_{i.j}) C_p \\ + P(Y_{i,j} < \mu_{i.j})(1 - C_p). \end{aligned} \tag{3.12}$$

Using (3.11), (3.12) and substituting in (3.6) we obtain the probability of binarization error

$$\begin{aligned} P_{B_e} = C_p \sum_{k=0}^{n} w_k \left( p_0 F(\alpha - t_k) - p_1 F(\beta - t_k) \right) \\ + (1 - C_p) \sum_{k=0}^{n} w_k (p_0 - p_1) F(Y_{i,j} - t_k) + p_1 \end{aligned}$$

where, $F$ is the cumulative distribution function of $f$ and $w_k = \binom{n}{k} p_1^k p_0^{n-k}$. The image error probability depends on the distribution of the transformation parameters $\alpha, \beta, C_p$ and the noise power level $\sigma_\eta$ in the QR code.

The expression for the embedded image, under above considerations, is given by:

$$Y_{i,j}^{out} = [\beta Q_{i,j} + \alpha(1 - Q_{i,j})] M_{C_p(i,j)} + Y_{i,j}(1 - M_{C_p(i,j)}). \tag{3.13}$$

The first component of this expression is the binary random variable $M_{C_p(i,j)}$ that selects the pixels to be modified and conserves the stochastic nature of the blue binary pattern. Despite the fact that the location of the modified pixels depends on the QR code binary pattern, it is reasonable to express them as independent Bernoulli random variables given that a sufficiently large window is considered. Furthermore, it is assumed an independence between QR code values, $Q_{i,j}$ and luminance values $Y_{i,j}$ in the original image. Finally the variable $Y_{i,j}$ is modeled as independent random variable following a local distribution, based on the size and location of the window in the image.

### *B. Error Detection Probability*

If it is assumed that central pixels of a module can be accurately sampled, the error probability can be expressed as:

$$\begin{aligned} P_{De} = \quad & P(\text{decide } Q_c = 1 | Q_c = 0) p_0 + \\ & P(\text{decide } Q_c = 0 | Q_c = 1) p_1 \end{aligned} \tag{3.14}$$

where, $Q_c$ are the central pixels value for a module and $p_0 = P(Q_c = 1)$ , $p_1 = P(Q_c = 1)$ are the probabilities of having the corresponding QR module inside the local window. In the above equation a slight difference with respect to the expression in (3.6) is presented. Where the probability of detecting an incorrect value only at the central position of the module is considered. This is a more relaxed requirement than the used in (3.6), since only the mean value of non-central pixels are considered meaning. This leads to a different probability model, where non detection probability in (3.14) is reduced to:

$$\begin{aligned} P(\text{decide } Q_c = 1 | Q_c = 0) = P(\alpha_c > t) \\ P(\text{decide } Q_c = 0 | Q_c = 1) = P(\beta_c < t). \end{aligned} \tag{3.15}$$

By considering the luminance transformation in (2.1) and by using the four luminance levels, the binarization thresholds at the reader are given by: $t = z + b + \eta$. Where, $z = \frac{\alpha n_\alpha + \beta n_\beta + \alpha_c n_{\alpha_c} + \beta_c n_{\beta_c}}{N}$, and $b$ is the mean value of unmodified pixels in the window, $n_\alpha, n_\beta, n_{\alpha_c}, n_{\beta_c}$ is the number of pixels with luminance level modified, $N$ is the total number of pixels, $b$ can be regarded as a Gaussian random variable with mean $\mu_b = E[Y_{1,j}](1 - C_p)$ and variance $\sigma_b^2 = \frac{Var[Y_{i,j}]}{N}(1 - C_p)$ independent of the noise $\eta$. Therefore the probability distribution of $t$ is also Gaussian with mean $\mu_t = z + \mu_b$ and variance $\sigma_t^2 = \sigma_\eta^2 + \sigma_b^2$. Its cdf can be expressed as function of the Gaussian normal cdf, $P(t < x) = \Phi\left(\frac{x - \mu_t}{\sigma_t}\right)$. To find the error detection probability is then reduced to combine the above result with the expressions in (3.7) and (3.14), what yields to:

$$P_{De} = \Phi\left(\frac{\alpha_c - \mu_t}{\sigma_t}\right) p_0 - \Phi\left(\frac{\beta_c - \mu_t}{\sigma_t}\right) p_1 + p_1. \tag{3.16}$$

### *C. Global Error Probability*

From the previous probability models a particular case is when only a fraction of the sampled pixels corresponds to center of the region with modified luminance $\alpha_c$ or $\beta_c$. However, since the probability of sampling error can be precomputed for a given module and center, it is possible to find the global error probability by considering the event of successful sampling of the center region. Then, the global error probability is given by

$$P_{err} = P_{Be}p_s + P_{De}(1 - p_s). \tag{3.17}$$

## 3.2.3 Optimization

### 3.2.3.1 *Quality Metrics*

Based on the previous probability model, it is possible to accurately determine the error probability for a given set of luminance parameters and to guarantee that the embedded image can be decoded, however, it is necessary to define appropriate metrics in order to quantify its visual quality. A well-known quality metrics to measure image similarity is the mean square error (MSE). It objectively quantifies the strength of the signal error, however, the visual quality is not always correlated. Thus, in order to evaluate the similarity between the original and embedded images, a variation of the metric presented in [30] is proposed. This metric involves the called Structural Similarity or SSIM index proposed in [31], where luminance, contrast and structure measurements are considered. The selection is based on the fact that blocks of the QR code should be less visible when viewed from a large distance, but at the same time the embedded image should preserve important details when it is viewed at closer distances. This metric considers both the halftone visibility and image structure, where the general form of this metric is given by

$$J(Y^{out}, Y) = w_l G(Y^{out}, Y) + w_h(1 - MSSIM(Y^{out}, Y)). \tag{3.18}$$

where the term $G(Y^{out}, Y)$ gives more importance to the low pass error between the embedded and original image, while the term $(1-MSSIM(Y^{out}, Y))$ measures the structure dissimilarity, when high frequency components and the overall structure of the image are considered. The weighting factors selected for all the simulations in this work are $w_l = 1$ and $w_h = 1/2$

according to [32]. The term $G$ includes a model of the HVS and the difference between the images is calculated in the $HSL$ color space, since color is an important component in the similarity perception.

One of the most used models for the HVS was proposed by Näsänen in [33] and it consists of an exponential model $H(f_r) = e^{-kf_r}$ where $f_r$ is the radial frequency in cycles/degree. Where, the corresponding discrete impulse response by considering the viewing distance $D$ in inches and the resolution $R$ in dots/inch is given by

$$h[m,n] = \frac{k}{2\pi} \frac{1}{\left(\frac{k}{2\pi} + (\frac{180m}{\pi RD})^2 + (\frac{180n}{\pi RD})^2\right)^{3/2}} \tag{3.19}$$

where the constant $k$ is determined by fitting empirical data from the model [33].

The general expression for $G$ is given by

$$G(Y^{out}, Y) = \frac{1}{N} ||h * (L^{out} - L)||_F \tag{3.20}$$

where $L^{out}$ and $L$ are the lightness components of the embedded and original images in the $HSL$ color space and $N$ is the number of pixels in the window. Only the $L$ component is here considered, because $S$ and $H$ are not changed during the color optimization. The filter $h$ is defined in (3.19) whose parameters define for this work are $D = 9$ inches and $R = 150$ dpi. By considering all the above requirements, the final expression to measure the distortion metric used in the optimization is then given by

$$\begin{aligned} J(\alpha, \beta, \alpha_c, \beta_c, C_p) = {} & \tfrac{1}{N} ||h * (L^{out} - L)||_F \\ & + \tfrac{1}{2}(1 - MSSIM(Y^{out}, Y)) \end{aligned} \tag{3.21}$$

### 3.2.4 Parameters Optimization

In order to take advantage of local correlations between the image luminance value and the QR code data, the optimization of the transformation parameters is independently performed by using local overlapping windows. In this implementation, the size of the windows is selected of $40 \times 40$ pixels centered around each $8 \times 8$ pixels image block. This corresponds to the window proportions established in (2.1). Based on experimental tests, a good trade off between robustness and quality for typical scanning distances is obtained.

A block diagram of the optimization process is depicted in figure 3.1. After the initial subdivision of the image, code and masks, optimization of each window is performed in parallel. The combination of the global luminance parameters is performed by low pass filtering the array of solutions and interpolating until size of the original image is reached.

The local optimizations are based on the quality metrics defined in (3.21) and the error probability model in (3.17) as

$$\begin{aligned} \underset{\alpha,\beta,\alpha_c,\beta_c,C_p}{\text{argmin}} \quad & J_i(\alpha,\beta,\alpha_c,\beta_c,C_p) \\ \text{subject to} \quad & P_{err} < P_{max,i} \end{aligned} \tag{3.22}$$

The optimizations input parameters are $P_{max}$ which determines the maximum admissible error probability and the size of the central region in pixels.

### *Non-linear optimization method*

The optimization problem in the equation (3.22) is solved by means of a logarithmic barrier method [34]. Solutions are found by solving the sequence of problems and when the regularization parameter $g$ have decreased considerably. The solver used is based on a sequential quadratic programing with the trust region method presented in [35] and implemented by the $fmincon$ function of MATLAB.

$$\begin{aligned} \underset{\alpha,\beta,\alpha_c,\beta_c,C_p,g}{\text{argmin}} \quad & J(\alpha,\beta,\alpha_c,\beta_c,C_p) + g\ \log(d) \\ \text{subject to} \quad & P_{err}(\alpha,C_p) - P_{max} + d = 0 \end{aligned} \tag{3.23}$$

# Chapter 4

# RESULTS AND ANALYSIS

This chapter describes both the functions used in the proposed algorithm and the results achieved by applying the algorithm to different images.

## 4.1 Algorithm Description

The algorithm proposed has three inputs: i) The QR code generated by using Zxing library developed in [19]. ii) The original image to be embedded into the QR code. iii) The halftone mask, developed with the algorithm in [23]. On other hand, several functions conform the algorithm developed, but in this chapter only the principal ones are presented in order to clarify the algorithm functioning. The first function used correspond to *"matchsizes"* developed to crop and resize the mask and original image, in order to match with the QR code size, because parallelism of calculations over the QR code.

**Algorithm 1** Embedding a QR code in a Color Image

**Require:** $(QR, Img, Mask)$
$[MaskFinder, MaskcMaskgray] = Modifymask(QR, Maskgray)$
$[ImgP, MaskP, QRP, MaskcP] = cutpatch(Img, Mask, QR, Maskcenters)$
$[\alpha_{opt}, \beta_{opt}, C_{popt}$, Perr]=optimize(ImgP,MaskP, MaskcP, QRP)
$Imgout = Applytranscode(Img, QR, \alpha_{opt}, \beta_{opt}, C_{popt}$,Maskgray)
**return** $[Imgout]$

The second function *"Modifymask"* is used to keep the function pattern. This function was developed because to maintain speed and to detect right position and alignment, the original patterns must be conserved. The patches of the input images are extracted by using local overlapping windows described by the equation 2.1. The third function *"cutpatch"* is developed to extract in parallel the patches from the original image, QR code symbol and halftone mask. Initial conditions for the parameters are defined and then the optimization process is performed to find the optimal parameters for each patch by means of the function *"optimize"*.

**Algorithm 2** Optimize - helper function

**Require:** $[ImgP, MaskP, MaskcP, QRP]$
$psf = fspecial('gaussian', 11, 1.5)$
$hg = fspecial('gaussian', 11, 1.5)$
$Pe = Perr(x, QRP, MaskP, MaskcP, ImgP,$
$Pe2 = Perr2(QRP, MaskP, ImgP$
$myconshort = myconspth(Pe, Pe2, Pth, p_s)$
$fidelity = @x \quad Jmodel(x, ImgP, L, QRP, MaskP, psf, hg, MaskcP)$
$cost = @x \quad fidelity(x)$
[$\alpha_{opt}$, $\beta_{opt}$, $C_{popt}$, $Perr$]= fmincon(cost,$\alpha$, $\beta$, $\alpha_c$, $\beta_c$,myconshort)
**return** $[\alpha_o pt, \beta_o pt_{Cpo} pt, Perr]$

By means of *"optimize"* function in algorithm 2, the probabilities and visual metrics are calculated in order to find the optimum values for $\alpha$, $\beta$ and the concentration of pixels to be changed. In particular, for all the figures in this work it was implemented the algorithm using the shelf nonlinear optimization function fmincon. Also, it was implanted the parallelism using the parallel computation toolbox in Matlab using 12 simultaneous workers taking an average of 30 minutes to process an image of $350 \times 350$ pixels in color. The processing time however does not depend heavily on the size of the image but on the number of blocks considered. The optimization of each block takes around 1 second for color images. Since, it was used a block size of $8 \times 8$ pixels it had to solve around 1900 optimizations per embedding. The optimization times also depend slightly in the contents of the image since the models of the probability of error and the image quality heavily depend on it. Finally, the optimal parameters are reassigned for the resultant image.

In order to have successful decoding the maximum probability of error $P_{max}$ used in the optimizations should be less than the error correction capacity of the code and leave a margin for other types of degradations such as uneven illumination or other disturbance that are not modeled. In all the examples in this work was used a QR code version 5-H which has a correction capacity of $30\%$ and therefore it was set the value of $P_{max}$ to $20\%$ leaving a margin for unexpected error.

## 4.2 Results

Embedding of different images into a QR code by using the algorithm proposed are presented in figure 4.1 and figure 4.3.

**Figure 4.1:** (Top Row)Three QR codes generated by the proposed method. (Bottom row) Original Images.

In order to evaluate the effectiveness of the embedding it was conducted several scanning experiments on a real setting. The images used for the experiments were previously classified by measuring its global complexity. The Global complexity metric measures the degree of complexity of an image based on images contrast and uniformity. There are several global metrics developed to measure image complexity, but this work is focus on the global gray-level uniformity metric defined in [36] as follows,

$$U = -\sum_{x}\sum_{y}[f(x,y) - \bar{f}(x,y)]^2 \tag{4.1}$$

where $f(x,y)$ is the gray level at pixel $(x,y)$ and $\bar{f}(x,y)$ is the average gray-level in a $3 \times 3$ window centered at $(x,y)$. In Figure 4.2 an image classification by its complexity degree is depicted. Big negative values of $U$, means that, more complex the images will be. The images were classified in three levels of complexity called high, medium and low. In figure 4.3 is presented a comparison of the results obtained with both the algorithm proposed and the algorithm developed by the company Visualed in [2]. The Visualed method was used for the comparison because this is the only on the market focused on visual improvement of QR codes by embedding images without using the redundant code information.

The resultant images of the algorithm proposed are visually pleasant and merge smoothly with

the QR code due to the use of halftone masks, while embeddings achieved with Visualed shows a pixeled appearance that severely decreased the visual content of the original image.


**Figure 4.2:** Image classification by complexity. The images are classified in high complexity (top), medium complexity (center), and low complexity images (bottom).


**Figure 4.3:** Image embedding obtained with the algorithm proposed and Visualed algorithm [2]

### *Experiment 1- Decoding Robustness*

This experiment was conducted to determine the robustness of the resultant embedding in real scanning environments, where external factors like uneven illumination, blur, among

others, are present. The experiment consist of scanning from a printed version and also from a laptop screen the images obtained with the algorithm proposed and the Visualed algorithm, the resultant embeddings are depicted in figure 4.3. The devices used were a smart phone Samsung Galaxy handset model version GT-S5830M and an ipad model A1395; for the scan it was used the mobile application *Barcode Scanner* based on Zxing library version 4.4.1. The HP LaserJet Pro 200 color Printer M251nw was used to print the embedding images of both algorithms. The results achieved are described in figures 4.4 and 4.5. The recognition times were in the scale of 2 to 9 seconds for printed versions of the embedding and from 1 to 4 seconds for a scan taken from laptop screen. The time achieved by scanning the embedding from a laptop screen is less than the time achieved in printed versions, probably because the illumination for printed versions is compensated by the screen-light in the laptop. A effective scanning depends on the resolution of the device and camera focus speed, most of the time spend in decoding is due to the camera trying to get in focus. It was found that the complexity of an image could affects the detection speed, taking more time for scanning embeddings with medium complexity. The experiment was applied for 10 repetitions of each image under normal illumination environment. The information contained in the resulting QR codes is readable for all the cases, however, printed version resolution and lighting conditions affect the decoding speed.

| Decoding results - Printed version | | | | | | |
|---|---|---|---|---|---|---|
| **High Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (f) - Baby | | Figure 4.3: (g) - Flag | | Figure 4.3: (h) - Ballons | |
| | Proposed Algorithm Figure 4.3: (f.1) | Visualed Algorithm Figure 4.3: (f.2) | Proposed Algorithm Figure 4.3: (g.1) | Visualed Algorithm Figure 4.3: (g.2) | Proposed Algorithm Figure 4.3: (h.1) | Visualed Algorithm Figure 4.3: (h.2) |
| **Ipad** | 3 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. |
| **Samsung** | 6 sec. | 2,1 sec. | 3 sec. | 3,4 sec. | 2 sec. | 1,7 sec. |
| **Medium Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (i) - Android | | Figure 4.3: (a) - Twitter | | Figure 4.3: (b) – Logo_Caracol | |
| | Proposed Algorithm Figure 4.3: (i.1) | Visualed Algorithm Figure 4.3: (i.2) | Proposed Algorithm Figure 4.3: (a.1) | Visualed Algorithm Figure 4.3: (a.2) | Proposed Algorithm Figure 4.3: (b.1) | Visualed Algorithm Figure 4.3: (b.2) |
| **Ipad** | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 3 sec. | 1 sec. |
| **Samsung** | 8 sec. | 3,2 sec. | 9 sec. | 3,2 sec. | 6 sec. | 2.1 sec. |
| **Low Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (c) - Skype | | Figure 4.3: (d) – Logo_UNAB | | Figure 4.3: (e) - Castle | |
| | Proposed Algorithm Figure 4.3: (c.1) | Visualed Algorithm Figure 4.3: (c.2) | Proposed Algorithm Figure 4.3: (d.1) | Visualed Algorithm Figure 4.3: (d.2) | Proposed Algorithm Figure 4.3: (e.1) | Visualed Algorithm Figure 4.3: (e.2) |
| **Ipad** | 1 sec. | 1 sec. | 1 sec. | 3 sec. | 1 sec. | 1 sec. |
| **Samsung** | 2 sec. | 1,1 sec. | 1 sec. | 2,2 sec | 1sec. | 1,9 sec. |

**Figure 4.4:** Decoding results. Process of scanning the Embedded QR code Images from printed versions.

| Decoding results – Laptop Screen | | | | | | |
|---|---|---|---|---|---|---|
| **High Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (f) - Baby | | Figure 4.3: (g) - Flag | | Figure 4.3: (h) - Ballons | |
| | Proposed Algorithm Figure 4.3: (f.1) | Visualed Algorithm Figure 4.3: (f.2) | Proposed Algorithm Figure 4.3: (g.1) | Visualed Algorithm Figure 4.3: (g.2) | Proposed Algorithm Figure 4.3: (h.1) | Visualed Algorithm Figure 4.3: (h.2) |
| **Ipad** | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. |
| **Samsung** | 3,6 sec. | 1,8 sec. | 2,2 sec. | 2,4 sec. | 2,5 sec. | 2 sec. |
| **Medium Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (i) - Android | | Figure 4.3: (a) - Twitter | | Figure 4.3: (b) – Logo_Caracol | |
| | Proposed Algorithm Figure 4.3: (i.1) | Visualed Algorithm Figure 4.3: (i.2) | Proposed Algorithm Figure 4.3: (a.1) | Visualed Algorithm Figure 4.3: (a.2) | Proposed Algorithm Figure 4.3: (b.1) | Visualed Algorithm Figure 4.3: (b.2) |
| **Ipad** | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. |
| **Samsung** | 3,8 sec. | 1,8 sec. | 2,9 sec. | 1 sec. | 3 sec. | 1,1 sec. |
| **Low Complexity Images** | | | | | | |
| **Image** | Figure 4.3: (c) - Skype | | Figure 4.3: (d) – Logo_UNAB | | Figure 4.3: (e) - Castle | |
| | Proposed Algorithm Figure 4.3: (c.1) | Visualed Algorithm Figure 4.3: (c.2) | Proposed Algorithm Figure 4.3: (d.1) | Visualed Algorithm Figure 4.3: (d.2) | Proposed Algorithm Figure 4.3: (e.1) | Visualed Algorithm Figure 4.3: (e.2) |
| **Ipad** | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. | 1 sec. |
| **Samsung** | 1,2 sec. | 1 sec. | 1,5 sec | 1,2 sec. | 1,4 sec. | 1,6 sec. |

**Figure 4.5:** Decoding results. Process of scanning the Embedded QR code Images from laptop screen versions.

### *Experiment 2 - Embeddings Visual Quality*

In this experiment, a comparison of the visual quality of the images between the original and the embedding is applied by using both the algorithm proposed and the developed by the company Visualed in [2]. The visual quality is measured for all the images in figure 4.3 by using the algorithm developed in [31], described by equation 4.2

$$SSIM(y^{out}, y) = \frac{(2\mu_{y^{out}}\mu_y + C_1)(2\sigma_{y^{out}y} + C_2)}{(\mu_{y^{out}}^2 + \mu_y^2 + C_1)(\sigma_{y^{out}}^2 + \sigma_y^2 + C_2)}$$

It was found that in general the algorithm proposed severely improves the quality of the embeddings while keeps decoding capacity. Best results were achieved for images with high complexity, the luminance changes are better disguised in an image with high amount of visual variety as background. The results of the comparison are described in figure 4.6

| **IMAGE SIMILARITY** | | | | | |
|---|---|---|---|---|---|
| **High Complexity Images** | | | | | |
| Figure 4.3: (f) - Baby | | Figure 4.3: (g) - Flag | | Figure 4.3: (h) - Ballons | |
| Proposed Algorithm Figure 4.3: (f.1) | Visualed Algorithm Figure 4.3: (f.2) | Proposed Algorithm Figure 4.3: (g.1) | Visualed Algorithm Figure 4.3: (g.2) | Proposed Algorithm Figure 4.3: (h.1) | Visualed Algorithm Figure 4.3: (h.2) |
| 64,06 % | 40,36 % | 99,36% | 61,47% | 99,51% | 62,30% |
| **Medium Complexity Images** | | | | | |
| Figure 4.3: (i) - Android | | Figure 4.3: (a) - Twitter | | Figure 4.3: (b) – Logo_Caracol | |
| Proposed Algorithm Figure 4.3: (i.1) | Visualed Algorithm Figure 4.3: (i.2) | Proposed Algorithm Figure 4.3: (a.1) | Visualed Algorithm Figure 4.3: (a.2) | Proposed Algorithm Figure 4.3: (b.1) | Visualed Algorithm Figure 4.3: (b.2) |
| 55,06% | 30,18% | 62,6% | 35,06% | 63,8% | 41,50 |
| **Low Complexity Image** | | | | | |
| Figure 4.3: (c) - Skype | | Figure 4.3: (d) – Logo_UNAB | | Figure 4.3: (e) - Castle | |
| Proposed Algorithm Figure 4.3: (c.1) | Visualed Algorithm Figure 4.3: (c.2) | Proposed Algorithm Figure 4.3: (d.1) | Visualed Algorithm Figure 4.3: (d.2) | Proposed Algorithm Figure 4.3: (e.1) | Visualed Algorithm Figure 4.3: (e.2) |
| 99,53% | 57,47% | 43,1% | 40,83% | 38,17% | 33,9% |

**Figure 4.6:** Structural Similarity comparison for the resultant embeddings achieved with the algorithm proposed and the algorithm develop by Visualed in [2]

# Chapter 5

# CONCLUSIONS

A method to automatically embed an image or logo into a QR code was proposed in this work. This allows to control the error detection probability and to embed any image into a QR code in such a way that most of the decoders can detect dark and light regions.

The halftone techniques used in this approach were adequate for printing the QR code on paper or rendering it in a digital screen. They reduce the pixeled appearance of the embedded image, furthermore, no other approaches at this point are based on halftoning algorithms.

An accurate approximation of the error probability was made by using a common binarization algorithm widely used in real applications. The main difference of the proposed method with respect to available solutions is that, the optimization is done by changing the luminance values in pixels selected by a halftone mask and specific values for the center of each module are defined.

The main drawbacks of the proposed method is the assumption that the pixels in the image are corrupted only by Gaussian noise.

Different experimental trials using mobile devices and printed codes have shown that for different scanning software applications available in the market and for common scanning distance, the decoding rates were reasonable in comparison with similar methods. However, a noticeable improvement of the QR code visual quality was obtained.

Finally, although the error probability was calculated by means of the binarization method proposed by Zxing library, which is not part of the QR code standard, it is possible that reading applications that use a different method could decode the resultant QR code.

# BIBLIOGRAPHY


[1] Russ Cox. Qart codes, April 2012, http://research.swtch.com/qart.

[2] Visualead Company. Qr code generator, april 2012, http://www.visualead.com/.

[3] RACO Industries. Raco barcode generator, October 2012, http://www.racoindustries.com/barcodegenerator/.

[4] Wakahara T. and Yamamoto N. Image processing of 2-dimensional barcode. *IEEE NBiS'11*, pages 484–490, 2011.

[5] Kuribayashi M. Fujita K. and Morii M. Expansion of image displayable area in design qr code and its applications. *Forum on Information technology*, 4:454–458, 2011.

[6] Floyd R. and Steinberg L. An adaptive algorithm for spatial gray-scale. *Proceedings Society Information Display*, 17(2):75–78, 1976.

[7] Ulichney R. The void-and-cluster method for dither array generation. *Proceedings SPIE*, 1913:332–343, 1993.

[8] Tan Jin Soon. Qr Code. *Synthesis Journal,Information Technology and Standards Committee Singapore*, 2008(3):59 – 78, 2008.

[9] QR code generator. Qr suite, Feb 2014, http://www.qr-code-generator.com.

[10] S. Ono and S. Nakayama. A system for decorating qr code with facial image based on interactive evolutionary computation and case-based reasoning. In *Second World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 401 – 406, Fukuoka, Japan, December 2010.

[11] International Organization for Standardization. Information technology, international symbology specification ï£¡ pdf417. ISO/IEC 15438:2006(E), International Organization for Standardization, Geneva, Switzerland, 2006.

[12] International Organization for Standardization. Information technology, international symbology specification - data matrix bar code symbology specification. ISO/IEC 16022:2006, International Organization for Standardization, Geneva, Switzerland, 2006.

[13] International Organization for Standardization. Information technology, international symbology specification ï£¡ maxicode. ISO/IEC 16023:2000, International Organization for Standardization, Geneva, Switzerland, 2000.

[14] International Organization for Standardization (ISO). Information technology automatic identification and data capture techniques barcode symbology qr code. ISO/IEC 18004:2000, International Organization for Standardization, Geneva, Switzerland, 2000.

[15] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[16] AD Brink and NE Pendock. Minimum cross-entropy threshold selection. *Pattern Recognition*, 29(1):179–188, 1996.

[17] J Zhou, Y Liu, and P Li. Research on Binarization of QR Code image. In *Multimedia Technology (ICMT), 2010 International Conference on*, pages 1–4, 2010.

[18] Jaakko Sauvola and Matti Pietikáinen. Adaptive document image binarization. *Pattern Recognition*, 33(2):225–236, 2000.

[19] Sean Owen. zxing, multi-format 1d/2d barcode image processing library with clients for android, java and c++, October 2012, https://code.google.com/p/zxing/.

[20] Morinaga K. Ono S. and Nakayama S. Two-dimensional barcode decoration based on real-coded genetic algorithm. *IEEE Congress on Evolutionary Computation*, pages 1068 – 1073, 2008.

[21] Wakahara T. and Yamamoto N. Image processing of dotted picture in the qr code of cellular phone. *IEEE 3GPCIC'10*, pages 454–458, 2010.

[22] AJ Gonzalez, GR Arce, J Bacca Rodriguez, and DL Lau. Human visual alpha-stable models for digital halftoning. In *18th Annual Symposium on Electronic Imaging Science and Technology: Human Vision and Electronic Imaging XI*, 2006.

[23] Gonzalo J. Garateguy, Gonzalo R. Arce, and Daniel L. Lau. Voronoi tessellated halftone masks. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 529–532. IEEE, 2010.

[24] Robert A Ulichney. Void-and-cluster method for dither array generation. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 332–343. International Society for Optics and Photonics, 1993.

[25] Daniel L Lau, Gonzalo R Arce, and Neal C Gallagher. Digital color halftoning with generalized error diffusion and multichannel green-noise masks. *Image Processing, IEEE Transactions on*, 9(5):923–935, 2000.

[26] Robert A Ulichney. Dithering with blue noise. *Proceedings of the IEEE*, 76(1):56–79, 1988.

[27] Analoui M. and Allebach J.P. Model based halftoning using direct binary search. *Proceedings SPIE*, 1666:96–108, 1992.

[28] Risto Nasanen. Blue-noise halftoning for hexagonal grids. *IEEE Trasn. on Image Proc*, 15(5):1270–1284, 2006.

[29] G.J. Garateguy, G.R. Arce, D. Lau and O.P. Villarreal. QR Images: Optimized Image Embedding in QR Codess. *Image Processing, IEEE Transactions on*, 23:2842 – 2853, 2014.

[30] T-T. Wong D. Cohen-Or W.-M. Pang, Y. Qu and P.-A. Heng. Structure-aware halftoning. *ACM Transaction on Graphics (TOG)*, 27(3):89, 2008.

[31] Sheikh H. Wang Z., Bovik A and Simoncelli E. Image quality assessment: From error measurement to structural similarity. *IEEE Transctions on Image Processing*, 13(1), 2004.

[32] Wong T. Cohen-Or D. Pang W., Qu Y. and Heng P. Structure-aware halftoning. *ACM Transactions on Graphics (TOG)*, 27(3):89, 2008.

[33] Risto Nasanen. Visibility of halftone dot textures. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):920–924, 1984.

[34] Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

[35] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[36] Richard Alan Peters, II, Richard Alan, Peters Ii, and Robin N. Strickland. Image complexity metrics for automatic target recognizers, 1990.