IMPROVING THE AESTHETIC APPEARANCE OF QR CODES

by

Ofelia P. Villarreal

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Summer 2014

© 2014 Ofelia P. Villarreal
All Rights Reserved
IMPROVING THE AESTHETIC APPEARANCE OF QR CODES

by

Ofelia P. Villarreal

Approved: _____________________________________________________________

Gonzalo R. Arce, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____________________________________________________________

Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____________________________________________________________

Babatunde Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____________________________________________________________

James G. Richards, Ph.D.
Vice Provost for Graduate and Professional Education
ACKNOWLEDGMENTS

I want to specially thank my advisor Gonzalo R. Arce. 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 Department of Electrical and Computer Engineering for providing their assistance and support during my master program.

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

My friend Gonzalo Garateguy for his collaboration, he was always available for my questions and gave generously of his time and knowledge on the path to this degree.

My family and friends, for their company in this part of my life.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. v
ABSTRACT ........................................................................ vii

Chapter

1 INTRODUCTION ................................................................. 1

2 BACKGROUND THEORY .................................................... 4

   2.1 Characteristics of QR codes ............................................ 4
   2.2 Halftone Techniques ..................................................... 5

      2.2.1 Error Diffusion ..................................................... 7
      2.2.2 Void and Cluster Dithering Arrays ......................... 8
      2.2.3 Direct Binary Search ........................................... 10
      2.2.4 Voronoi Tessellated Halftone Masks ..................... 12

3 BINARIZATION AND LUMINANCE MODIFICATION .............. 14

   3.1 Luminance Modification ............................................. 18

4 PROBABILITY OF ERROR AND OPTIMAL SELECTION OF
   PARAMETERS ............................................................... 20

5 IMAGE EMBEDDING RESULTS ........................................... 26

6 CONCLUSIONS ................................................................. 30

BIBLIOGRAPHY .................................................................... 31

Appendix

   DISCLAIMER .................................................................. 33
# LIST OF FIGURES

1.1 QR features scheme. .................................................. 3

2.1 Halftone technique applied to grayscale image. ................ 6

2.2 Error Diffusion algorithm in [1]. .................................. 7

2.3 Arrangement result of the Void and Cluster method using a two-dimensional Gaussian filter for a 16 x 16 input pattern obtained in [2]. .................................................. 9

2.4 a) Void and Cluster method for halftoning Lena image. b) Error Diffusion for halftoning the Lena image made in [2]. .............. 10

2.5 DBS algorithm for processing of each pixel. ..................... 11

2.6 Voronoi tessellations patterns achieved with different algorithms [3]. (a) Void and cluster technique. (b) Direct Binary Search. (c) Lloyd algorithm. .................................................. 13

2.7 Halftone mask obtained by using Voronoi Tessellation. ........ 13

3.1 Blocks subdivision in the threshold calculation algorithm used by Zxing library. .................................................. 15

3.2 Binarization results for different thresholding methods. a) Original image. b) Binary image using the midgray threshold. c) Otsu’s method. d) Local thresholding with Zxing library. ..................... 17

4.1 a) Probability of binarization error for different values of $p_c$. The vertical lines correspond to the transition threshold of (4.14). b) The shaded region correspond to the region below the bound (4.14). c) Color patch transformed with the pair of parameter marked in b), the (upper left) Figure is the original image, (upper right) is the QR code, (lower left) is the embedded code and (lower right) the binarization of this code. .... 24

5.1 (Top row) Four QR codes generated by our method. (Bottom row) Original Images. .................................................. 26
5.2 (Top row) Variation of the \( p_c \) parameter to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for \( \alpha^* \). ................................................. 27

5.3 (Top row) Variation of the \( \lambda \) parameter to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for \( \alpha^* \). ................................................. 28

5.4 (Top row) Variation of \( p_c \) and \( \lambda \) parameters to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for \( \alpha^* \). ................................................. 29

vi
ABSTRACT

This work introduces a method to improve the aesthetic appearance of QR codes subject to bounded probability of detection error. The resultant QR code image embedding is compatible with standard decoding applications for mobile devices available in the market. This method takes advantage of the immunity of QR readers against local luminance disturbances and intends to render the colors of a logo in such a way that the luminance in the dark regions of the codes is detected as dark and the light regions as light. This approach exploits the rich theory and practice of digital halftoning for pixel selection to optimally fuse images into QR codes. The technique minimizes the dual criteria of visibility and decodability. A tractable model for the probability of error is developed. The robustness of the method is adjustable by changing a set of parameters in order to handle variations in the tone given by poor illumination or other factors. The error correction capacity is given by the correction capacity of the QR code. Experimental results are presented evidencing considerable visual quality improvement.
Chapter 1
INTRODUCTION

QR codes (Quick response Codes), were developed in 1994 by Denso corporation [4] and have rapidly emerged as a widely used inventory tracking and identification method in transport, manufacturing, and retail industries. These codes are extensively used today in marketing and publicity to provide enhanced information or augmented reality experiences. Their popularity is due in part to the proliferation of smart phones, capable of quickly decoding the information and access websites or download online content. QR codes are sufficiently robust to be decoded without special illumination or cameras, making them very popular to distribute, contact information or URLs. This lead to their adoption in 1999 as a Japanese industrial standard (JIS-X0510) and eventually in 2000 as an International Standard (ISO/IEC18004) [5]. QR codes can be considered as the evolution of other two dimensional codes like PDF147 and MAXI codes which allow for high capacity and high decoding speeds respectively [6, 7]. QR codes area designed to inherit these characteristics and to improve the ease of use. Their decoding speed is based on the use of finder and alignment patterns which allow to rapidly detect and correct orientation and geometric distortions. The finder patterns are located in three of the four corners of the symbol and the alignment patterns are inside the symbol to help in the correction of uneven or irregular surface distortion (see Figure 1.1). Another important characteristic that contributed to the popularization of QR codes is their efficiency in encoding Kanji and Kana Japanese symbol characters. Factors like uneven illumination conditions are not taken into account in the standard, and the handling of this process is left to the implementation [5]. This stage however is key to the robustness of the whole decoding algorithm and it become specially relevant in the cases where logos or images are superimposed to the QR codes. There have been numerous efforts to improve the appearance of QR codes by embedding logos and images into them or by changing their color patterns in
order to generate visually appealing codes [8, 9]. This deviation from the standard come at the cost of robustness with respect to the different distortions and should be performed with care to maintain decodability. One of the approaches used is to replace some of the information modules with the desired logo or image, and rely on the error correcting capabilities of the code to maintain decodability [10]. Other approaches more visually appealing but computationally complex used luminance modification to embed images into the QR code [11, 12]. Some approaches however are based on trial an error and require expertise to yield successful results. The main challenge of blending images with QR codes is that the decoder cannot be altered, otherwise it will no longer be standard compliant. The simplicity of this decoding algorithm is a key feature that allow QR codes to be fast and suitable to be implemented in smart phones or other embedded devices. However the simple thresholding used to binarize the codes imposes severe limitations in the amount of degradation that the code can withstand while keeping decodability [5]. In this work we present a simple yet effective algorithm to transform the luminance of an image according the desired QR code. Some of the results achieved were previously published in [12], as a result of the joint work developed. In this work, the luminance values of a percentage of the pixels of a QR code are modified. the modified pixels are selected according to the structure of a halftoning mask. This selection method permits to preserve the high frequencies in the analyzed images. The probability of error in the binarization of the QR code is also calculated. The embedded image quality is optimized using the binarization probability error of the QR code and the mean square error between the input and the embedded images. The algorithm proposed to modify the image is rather simple and only has two tunable parameters; the number of modified pixels per module Nc and the luminance levels \(\alpha, (1 - \alpha)\) to which these pixels are transformed. This approach also exploits the rich theory and practice of digital halftoning to optimally fuse images into QR codes, minimizing the dual criteria of visibility and decodability. Finally we experimentally calculate the probability of correct decoding for both the black and white and the embedded code showing that their decoding performances are very similar while the embedded code has the advantage of preserving much of the visual information in the image.

The organization of the work is as follows. Chapter II presents the structure
and main features of QR codes and halftoning background. Chapter III describes the most common binarization procedures used by public available implementations and the luminance manipulation algorithm. Chapter IV describes the probability of error in the binarization process as a function of the algorithm parameters. Chapter V presents the results of applying this algorithm to several images and logos and finally Chapter VI presents some conclusions and future work.
Chapter 2

BACKGROUND THEORY

2.1 Characteristics of QR codes

QR codes have a particular structure defined by its multiple patterns and the organization of the information bits. The symbol structure can be divided in two sections, function patterns and encoding region.

A. Function Patterns

Function patterns contain the finder and alignment patterns necessary to locate, position and align the symbol. The three main patterns are finder patterns, alignment patterns and timing patterns. The function of finder patterns is to speed up the alignment and detection of the code. By using this patterns the speed to identify and decode the symbol increases by a factor of 20 in comparison with other matrix codes. The finder patterns are big squares located in three of the four corners of the code, having a dark and light distribution ratio of 1 : 1 : 3 : 1 : 1 between scan lines (see Figure 1.1). Alignment patterns are used to correct for perspective transformations and consist of a series of enclosed squares of 5 × 5 dark modules, 3 × 3 light modules and a single dark module in the center as depicted in Figure 1.1. The number of alignment patterns depends on the symbol version and the distribution is made as uniform as possible through the encoding area. Finally timing patterns are used to determine the modules coordinates in the symbol and consist of a row and a column following an alternate arrangement of dark and light modules through the symbol. This patterns help to locate the central coordinate of the modules, which are very useful when the symbol is distorted. The standard also defines a quite zone which is represented by the white border around the QR code. This zone helps to improve the detection of the code by separating the symbol from other images or marks in its surroundings. The width of the quiet zone must be at least four
modules and its nominal reflectance must be the same as lightest module in the symbol [5].

B. Encoding Region

The encoding region is the remaining space after incorporating all the function patterns. This region is used to store the data, error correction information, version and format information of the QR code. Any type of data to be encoded must be converted into a bit string in agreement with the encoding rules. The resultant bit stream is fragmented into 8-bit codewords (dark and white modules) which are arranged in this region.

C. Size, Data Capacity and Version Information

QR codes are able to store a considerable amount of data in comparison with previous types of codes. However, the store capacity varies and is depending on the version, error correction level and the type of characters. It is consider that for numeric data only, the QR code can store 7089 characters, 4296 alphanumeric characters, 2953 of byte characters or 1817 kanji/kana characters. QR codes have forty different versions that goes from $21 \times 21$ modules (first version) to $177 \times 177$ modules (last version). The redundant information required for error correction is included in this modules as well as the alignment patterns. Because of this their information capacity is slightly smaller.

D. Error Correction

QR codes incorporate error correction to improve reliability in the presence of partial damage and noise in the symbol. There are four levels of error correction L, M, Q, H. Each level corresponds to a percentage of resistance to damage of 7%, 15%, 20%, 30%, respectively. The amount of modules added to the data modules is proportional to the correction level. The block code used to encode the information bits is Reed Solomon.

2.2 Halftone Techniques

The embedding Images and logos into QR codes are constrained printing optimization problems. Digital halftoning simulates continuous tone imagery through the use of digitally printed dots, varying either in size or in spacing (see Figure 2.1). While continuous tone images contain an infinite range of colors or grays, halftone reproductions
only use a finite set of colors for dot printing. This produces an image quantization error which is naturally mitigated by the characteristics of the human eye, which blends the coarsely quantized print into smoother and visually pleasing tones. Halftoning for QR code embedding, in addition, adds the constraint that the luminance of the printed dot patterns, when viewed by the QR scanner, must reproduce the binary QR code cover fairly well. In other words, the halftoning algorithm is designed such that the color to luminance transformation in the QR code readers produces an image that is as close as possible to a scaled replication of the original QR code. Among the algorithms proposed in Halftoning literature to create halftone images are, error diffusion, develop by Floyd and Steinberg [1], Void and Cluster [2], Direct Binary Search[13], and the most representative are techniques based on Centroidal Voronoi Tessellation [3].

![Gray scale image](image1.png) ![Halftone generated by error diffusion](image2.png)

**Figure 2.1:** Halftone technique applied to grayscale image.
2.2.1 Error Diffusion

This approach was developed by [1] 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 2.2 and described by:

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

(2.1)

where,

\[
w^e[n] = \sum_{i=1}^{M} c_i \cdot v^e[n - i],
\]

(2.2)

where \(c\) is the error filter develop by Floyd and Steinberg [1] defined as,

\[
c = \begin{bmatrix}
- & # & 7/16 \\
3/16 & 5/16 & 1/16
\end{bmatrix}.
\]

(2.3)

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 processes 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).

![Error Diffusion algorithm](image)

**Figure 2.2:** Error Diffusion algorithm in [1].
2.2.2 Void and Cluster Dithering Arrays

Void and Cluster refers to the arrangement of minority pixels on the background of majority pixels, in this way void refers to a large physical space between minority pixels while cluster refers to close grouping of majority pixels. Accomplishing an homogeneous distribution of 0s and 1s implies to 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 create two matrices in order to improve the ranking process. The first matrix is a $M \times N$ matrix, called Dither array. The second matrix is denoted as Prototype Binary Pattern (PBP) by [2] and has the same size as the dithering matrix which helps in the ranking process. The rank includes the values between 0 and $(MN - 1)$. Each element in the dithering array corresponds to the number of 1s that exists in the PBP less the 1 in the location of the element. In this way, the creation of a Dither array involves the simultaneous processing of two matrices (Dithering array and Prototype Binary Pattern). Having into account that minority pixels correspond to less than half of the pixels in PBP that are 1s and majority pixels are the remaining 0s pixels. Ulrichney [2] proposed the use of a filter 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 [2] by the following equation (2.7),

$$D(x, y) = \sum_{r=-M/2}^{M/2} \sum_{s=-M/2}^{M/2} PBP(r', s')f(r, s),$$  \hspace{1cm} (2.4)

where,

$$r' = (M + x - r) \text{module } M,$$  \hspace{1cm} (2.5)

$$s' = (M + x - s) \text{module } M,$$  \hspace{1cm} (2.6)

and where $D(x, y)$ 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

$$f(r, s) = e^{-\frac{r^2}{2\sigma^2} - \frac{s^2}{2\sigma^2}}.$$  \hspace{1cm} (2.7)
The arrangement results over a $16 \times 16$ input pattern are depicted, in Figure 2.3.

\begin{figure}[h]
\centering
\includegraphics[width=0.7\textwidth]{figure2_3.png}
\caption{Arrangement result of the Void and Cluster method using a two-dimensional Gaussian filter for a $16 \times 16$ input pattern obtained in [2].}
\end{figure}

In Figure 2.4 (a), (b) a comparison between Void and Cluster and Error Diffusion method is shown. Void and Cluster method obtains good results in halftoning images, however, due to the sharpening properties of Error Diffusion it achieves better results.
Figure 2.4: a) Void and Cluster method for halftoning Lena image. b) Error Diffusion for halftoning the Lena image made in [2].

2.2.3 Direct Binary Search

This technique attempts to visually characterize optimal patterns by incorporating a Human Visual System model which will decide what dot must be keep in the halftone image to represent accurately the original continuous-tone. In this way, beginning with a halftone image, the algorithm process each pixels at a time. Having into account a $3 \times 3$ window, the current pixel can be swapped by one of the pixels in the window, or can be toggled from 1 to 0 or viceversa. According to the model developed, as shown in Figure 2.5, integrating a model of the Human Visual System (HSV) defined by $h(x)$ a continuous space linear filter, the modeled output is defined in [13] as

$$\tilde{z}(x) = z(x) * h(x) = \sum_{n} g[n] \tilde{p}(x - Xn),$$  \hspace{1cm} (2.8)
**Figure 2.5:** DBS algorithm for processing of each pixel.

where,

\[ z(x) = \sum_n z[n]p(x - Xn), \]  
\[ (2.9) \]

and

\[ \tilde{p}(x) = p(x) * h(x), \]  
\[ (2.10) \]

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

\[ f(x) = \sum_n f[n]p(x - Xn), \]  
\[ (2.11) \]

and

\[ \tilde{f}(x) = \sum_n f[n]\tilde{p}(x - Xn). \]  
\[ (2.12) \]

The error is given then by

\[ \tilde{e}(x) = \tilde{f}(x) - \tilde{z}(x). \]  
\[ (2.13) \]

Swapping or toggling pixels in the \( 3 \times 3 \) window are evaluated having into account the overall error measured in equation (2.13). This iterative process ends when the error cannot be reduced in spite of the changes. This process requires significant computational
resources given that the halftone image must be recalculated for each swap or toggle, for each pixel of the image, in all iterations.

2.2.4 Voronoi Tessellated Halftone Masks

In a good binary pattern according to [14] minority pixels must have an aperiodic and isotropic distribution and should not contain any low frequency spectral components. Also the minimum distance between neighboring minority pixels must be defined according to (2.14).

$$\lambda_p = \begin{cases} 
\frac{1}{\sqrt{g'}}, & \text{for } 0 < g' \leq \frac{1}{4} \\
2, & \text{for } \frac{1}{4} < g' \leq \frac{3}{4} \\
\frac{1}{\sqrt{1-g'}}, & \text{for } \frac{3}{4} < g' \leq 1.
\end{cases} \quad (2.14)$$

This technique was first introduced in [3] to generate the mask. This technique perform a global optimization of the minority pixels by using Voronoi Tessellations at each step of the mask generation. Particularly, Lloyd algorithm generate Voronoi Tessellation patterns used to determine the location for the point to optimally satisfy the $\lambda$ constrain. A set of stacked binary patterns must be created; to recover a particular binary pattern by thresholding the mask, furthermore in order to accurately represents a given gray level, the set of white pixels must have a concentration such that $E[I_g[n]] = g/G - 1$. Voronoi Tessellation is calculated for each binary pattern having into account the optimality criteria defined by equation (2.14). 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 2.7) is defined as follows,

$$S = \left(\frac{G-2}{2}\right)1_{M \times N} + B - W, \quad (2.15)$$

where $B$ and $W$ are defined as:

$$B = \sum_{g=1}^{(G-2)/2} I_g^b, \quad (2.16)$$

$$W = \sum_{g=1}^{(G-2)/2} I_g^w. \quad (2.17)$$
**Figure 2.6:** Voronoi tessellations patterns achieved with different algorithms [3]. (a) Void and cluster technique. (b) Direct Binary Search. (c) Lloyd algorithm.

**Figure 2.7:** Halftone mask obtained by using Voronoi Tessellation.

The resultant mask achieves a better approach than the obtained with Void and Cluster and Direct Binary Search. In Figure 2.7 a mask obtained with Centroidal Voronoi Tessellation is depicted.
Chapter 3
BINARIZATION AND LUMINANCE MODIFICATION

A. Binarization

The binarization stage is the first process in the QR decoding sequence. The rest of the operations including the alignment, orientation detection and rectification are performed over the binary image. Therefore it is very important to segment the gray scale image as good as possible since a failure in this stage results in no decodability. The segmentation task is particularly challenging when there are variations in the lighting conditions. The standard suggest a very simple rule consisting in the use of the global mid gray threshold, defined by $t = \frac{\max(Y) + \min(Y)}{2}$ where $Y$ is the luminance image. Among global thresholding techniques Otsu’s method [15] is one of the most effective when the global histogram of the image is approximately bimodal. This method have been applied successfully for QR binarization, and it is possible to implement very efficiently. In the case of uneven illumination or when noise levels are high, global thresholding techniques have a very poor performance, since the histogram of the image is not bimodal, and any technique based only on the values of the pixels will miss classify some regions. A better approach in these cases is the use of local thresholding techniques in which the thresholds are calculated based on a running window. Several functions have been proposed such as, mid gray (3.1), median (3.2) or mean (3.3),

$$t_{i,j} = \frac{\max(Y_{Bi,j}) + \min(Y_{Bi,j})}{2},$$ \hspace{1cm} (3.1)

$$t_{i,j} = \text{median}(Y_{Bi,j}), \hspace{1cm} (3.2)$$

$$t_{i,j} = \text{mean}(Y_{Bi,j}), \hspace{1cm} (3.3)$$
Figure 3.1: Blocks subdivision in the threshold calculation algorithm used by Zxing library.

where $Y_{B_{i,j}}$ is a window centered around the $[i, j]$ pixel. The rest of this thesis focuses on the binarization method used in the Zxing library [16] which uses local histograms to calculate the thresholds. The first stage in the threshold calculation algorithms divides the image in blocks $B_{m,n}$ of $8 \times 8$ pixels and calculates the average of the luminance values for each block,

$$
\mu_{m,n} = \frac{1}{64} \sum_{(i,j) \in B_{m,n}} Y[i,j].
$$

(3.4)

There are exceptions in the library to manage situations when the luminance variation in $B_{m,n}$ is very small, the threshold may forced to 0 or calculated based on neighboring modules. However, for the purpose of the theoretical analysis, in this work we will assume that all $\mu_{m,n}$ are calculated as in equation (3.4). After calculating the block averages, the threshold for each pixel is obtained according to algorithm 1.
Algorithm 1 Threshold Calculation Algorithm

Require: $\mu_{m,n}$ for $m = 1, \ldots, [M/8]$ and $n = 1, \ldots, [N/8]$
1: $i \leftarrow 1, j \leftarrow 1$
2: for $i = 1$ to $[M/8]$ do
3:   for $j = 1$ to $[N/8]$ do
4:     Select $m, n$ such that $(i, j) \in B_{m,n}$
5:     if $m \leq 2$ then
6:       $k = 2$
7:     end if
8:     if $m > 2$ and $m \leq [M/8] - 2$ then
9:       $k = m$
10:    end if
11:   if $m > [M/8] - 2$ then
12:     $k = [M/8] - 2$
13:   end if
14:   if $n \leq 2$ then
15:     $l = 2$
16:   end if
17:   if $n > 2$ and $n \leq [N/8] - 2$ then
18:     $l = n$
19:   end if
20:   if $n > [N/8] - 2$ then
21:     $l = [N/8] - 2$
22: end if
23: $t_{i,j} = \frac{1}{25 \times 64} \sum_{p=k}^{p=k+2} \sum_{q=l}^{q=l+2} \sum_{p-m-2}^{p-m+2} \sum_{q-n-2}^{q-n+2} \mu_{p,q}$, for $(i, j) \in B_{m,n}$
24: end for
25: end for

This algorithm performs an average of the $\mu_{m,n}$ values in a $5 \times 5$ window of blocks. The first part of the algorithm manages the edge blocks but in general for a central block of the image (see Figure 3.1) the threshold used for binarization is

$$t_{i,j} = \frac{1}{25 \times 64} \sum_{p=k}^{p=k+2} \sum_{q=l}^{q=l+2} \sum_{p-m-2}^{p-m+2} \sum_{q-n-2}^{q-n+2} \mu_{p,q}$$ for $(i, j) \in B_{m,n}$ \hspace{1cm} (3.5)

Combining this with equation (3.4), we obtain the general expression for the threshold in a central pixel of the image

$$t_{i,j} = \frac{1}{25 \times 64} \sum_{p=k}^{p=k+2} \sum_{q=l}^{q=l+2} \sum_{p-m-2}^{p-m+2} \sum_{q-n-2}^{q-n+2} Y[k, l]$$, for $(i, j) \in B_{m,n}$ \hspace{1cm} (3.6)
**Figure 3.2**: Binarization results for different thresholding methods. a) Original image. b) Binary image using the midgray threshold. c) Otsu’s method. d) Local thresholding with Zxing library.

The binary image $Q$ is obtained by thresholding the luminance image with $t_{i,j}$,

$$Q[i, j] = \begin{cases} 
1 & \text{if } Y[i, j] > t_{i,j} \\
0 & \text{if } Y[i, j] < t_{i,j}
\end{cases} \quad (3.7)$$

Figure 3.2 shows an example of the binary output using a mid gray threshold, Otsu’s method and the Zxing library in which the advantage of using local threshold for uneven illumination is clear.

**B. Decoding**

Decoding process follows the same steps applied in encoding procedure, but two additional layers are added. i) Binarization: The image is acquired by the smart phone’s camera in grayscale and then segmented in black and white pixels. ii) Detecting the right orientation of the symbol and the misaligned codewords. After these two steps, the decoding procedure follows the same steps as encoding but in reverse order.
3.1 Luminance Modification

A. Pixel Selection

In order to yield a smooth transition between the code and the image, the spatial
distribution of modified pixels should be as uniform as possible. One of the most efficient
ways to generate random but spatially uniform distribution of points is by means of a Blue
Noise masks. These masks have the characteristic that when compared with a constant
value generate a binary pattern with equidistant distributions of points. The inter point
distance \( \lambda \) between two closest points is determined by the threshold \( p_c \) as
\[ \lambda = \frac{1}{\sqrt{p_c}} \]
for \( 0 \leq p_c \leq 0.5 \) and
\[ \lambda = \frac{1}{\sqrt{1-p_c}} \]
for \( 0.5 < p_c \leq 1 \) [17]. These binary patterns approximate
a constant gray level by taking advantage of the low pass characteristics of the human
visual system. All patterns generated by thresholding a blue noise mask concentrate most
of their energy in the high frequencies and their mean value equals the threshold \( p_c \). For
the purpose of this luminance modification algorithm, the gray level rendered by the blue
noise pattern corresponds to the concentration of modified pixels \( p_c \). The set of modified
pixels is denoted by \( C \) and its concentration \( p_c = \frac{N_c}{W^2} \) is the ratio between the total number
of pixels in a QR module \( W \times W \) and the number of modified pixels per module \( N_c \). When
the concentration of modified pixels \( p_c \) approaches 1, the embedding resembles the QR
code while for lower concentrations it is closer to the original image.

B. Luminance Modification Algorithm

The first step in the algorithm is to select a subset of pixels in the image. We select
this subset following a blue noise distribution in order to minimize the visual impact of
the modification [18, 19]. Lets consider the image pixels corresponding to one module of
the QR code. We denote by \( C \) the set of \( N_c \) pixels to be modified in each module. Each
module of the code consist of a square of \( M \times M \) pixels and the probability of selecting
a modified pixel in the module is \( p_c = \frac{N_c}{M^2} \). After selecting the subset of pixels we change
its luminance value \( Y[i, j] \) according to the following rule

\[
Y^{out}[i, j] = \begin{cases} 
1 - \alpha & \text{if} (i, j) \in C, q_{i,j} = 1 \\
\alpha & \text{if} (i, j) \in C, q_{i,j} = 0 \\
Y[i, j] & \text{otherwise.}
\end{cases}
\]  

(3.8)
Note that by setting $N_c = 0$ we obtain the original image and by setting $N_c = M^2, \alpha = 0$ the binary code. In order to preserve the color information of each pixel, we change its luminance $Y$ by manipulating the color in the HSL color space. Other color spaces such as Lab can be used, but for the sake of speed and simplicity we perform all the manipulations in the HSL space in this thesis. The definition of luminance used throughout this work is $Y = 0.2989R + 0.5870G + 0.1140B$. To change $Y$ we first convert the RGB pixel to the HSL space, $[I, S, L] = T(R, G, B)$ and then modify the $L$ component, keeping $H$ and $S$ fixed in order to reach the target $Y$. By fixing $H$ and $S$ we can write $R, G$ and $B$ as a function of $L$ given by $[R(L), G(L), B(L)] = T^{-1}(H, S, L)$. Defining $w = [0.29890.58700.1140]^T$ we have that

$$Y(L) = f(L) = w^T T^{-1}(H, S, L), \quad (3.9)$$

where $L = \frac{\min(R, G, B) + \max(R, G, B)}{2}$ component in the HSL color space is a piecewise, linear and monotonic function $Y = f(L)$.

For a given luminance target $Y = \alpha$, the corresponding value of $L$ is obtained as $L^* = f^{-1}(\alpha)$. The algorithm to change the RGB colors of a pixel in order to meet the luminance target is described in algorithm 2.

---

**Algorithm 2** Color Modification for Luminance Target

**Require:** $(R, G, B)$ pixel and target value of $Y = \alpha$

1. $(I, S, L) \leftarrow T(R, G, B)$
2. find $L^* = f^{-1}(\alpha)$
3. $(R^*, G^*, B^*) \leftarrow T^{-1}(I, S, L^*)$
4. return $(R^*, G^*, B^*)$
Chapter 4

PROBABILITY OF ERROR AND OPTIMAL SELECTION OF PARAMETERS

Image embedded codes are a combination of pixels from the original image and the QR code. The mixture affects the binarization thresholds increasing the probability of binarization error. To avoid this situation the luminance targets $\alpha$ and $1-\alpha$ should be optimized to minimize the probability of binarization error. Consider the pixel at $(i, j)$, the probability of error is given by,

$$ P_{err} = P(\text{decide } q_{i,j} = 1|q_{i,j} = 0)P(q_{i,j} = 0) + P(\text{define } q_{i,j} = 0|q_{i,j} = 1)P(q_{i,j} = 1). \quad (4.1) $$

The probability of miss decoding a pixel depend on the binarization method used can be expressed as a function of the threshold and modified image values,

$$ P(\text{decide } q_{i,j} = 1|q_{i,j} = 0) = P(Y_{i,j}^{out} > t_{i,j}|q_{i,j} = 0) \quad (4.2) $$

$$ P(\text{select } q_{i,j} = 0|q_{i,j} = 1) = P(Y_{i,j}^{out} < t_{i,j}|q_{i,j} = 1). $$

$Y^{out}$ and $t_{i,j}$ are dependent random variables but the calculation of these probabilities can be simplified by explicitly including the threshold calculation method from (3.8) and doing some assumptions about the components of these thresholds. The modified image $Y$ in (3.8) is expressed as the combination of several random variables given by,

$$ Y^{out}[i, j] = [(1-\alpha)q_{i,j} + \alpha(1-q_{i,j})]x_{i,j} + y_{i,j}(1-x_{i,j}). \quad (4.3) $$

where $x_{i,j}$ is a Bernoulli random variable, such that $x_{i,j} = 1$ for $(i, j) \in C$. The first assumption concerns $x_{i,j}$, which is 1 for $(i, j) \in C$ and 0 otherwise. This variable is modeled as one independent Bernoulli random variable with probability $p_c$ equal to the concentration of modified pixels. Then the values of the QR code $q_{i,j}$ are also considered as
Bernoulli random variables independent from $x_{i,j}$ and from each other. Finally the luminance values of the original image $Y_{i,j}$ are modeled as random variables, independent form $q_{i,j}$ and $x_{i,j}$ whose probability density function $f_Y(y)$, is given by the local distribution of the patch.

Considering the mean block threshold calculation method in 3.6, it is possible to divide this threshold in two components $t_{i,j} = a_{i,j} + b_{i,j}$, where $a_{i,j}$ corresponds to the modified pixels in $C$ and $b_{i,j}$ to the unmodified luminance pixels,

$$
a_{i,j} = \frac{1}{25 \times 64} \sum_{p,q} \sum_{(k,l) \in B_{p,q}^C} (1 - \alpha) q_{k,l} + \alpha(1 - q_{k,l})$$

$$b_{i,j} = \frac{1}{25 \times 64} \sum_{p,q} \sum_{(k,l) \notin B_{p,q}^C} Y[k, l]. \tag{4.4}$$

Using the assumption that $q_{i,j}$ are independent Bernoulli random variables with $P(q_{i,j} = 1) = p_1$ and $P(q_{i,j} = 0) = p_0$ the distribution of $a_{i,j}$ is given by

$$P(a_{i,j} = x) = g(x) = \sum_{k=0}^{n} \binom{n}{k} \left(1 - \alpha\right)^k \alpha^{n-k} p_1^k p_0^{n-k}, \tag{4.5}$$

where $k$ is the number of pixels with modified luminance $1 - \alpha$ in the 2-neighborhood of $B_{m,n}$ and $n$ is the total number of modified pixels in the neighborhood, $n = |p_c(64 \times 25)|$. The distribution of $b_{i,j}$ depends on the position in the image but can be estimated from the local histogram. 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 and we denote this distribution by $P(b_{i,j} = y) = f(y)$. Since the values of $a_{i,j}$ and $b_{i,j}$ are also independent given $C$ then,

$$P(t_{i,j} = t) = (g * f)(t) = \sum_{k=0}^{n} f(t - t_k) \binom{n}{k} p_1^k p_0^{n-k}, \tag{4.6}$$

where $t_k = \left[\frac{k(1-\alpha)}{25 \times 64} + \frac{(n-k)\alpha}{25 \times 64}\right]$ are the discrete values obtained from the sum of $n$ realizations of $q_{i,j}$. In order to calculate the probabilities on (4.2) we need $P(t_{i,j} \leq T)$, which is calculated as,

$$P(t_{i,j} \leq T) = \sum_{k=0}^{n} \binom{n}{k} p_1^k p_0^{n-k} \int_0^T f(t - t_k) dt$$

$$= \sum_{k=0}^{n} \binom{n}{k} p_1^k p_0^{n-k} F(T - t_k). \tag{4.7}$$
The conditional probabilities on (4.2) are obtained by conditioning on \( C \) as follows,

\[
P(Y_{i,j}^{\text{out}} > t_{i,j}|q_{i,j} = 0) = P(\alpha > t_{i,j})p_c + P(Y_{i,j} > t_{i,j})(1 - p_c)
\]  

(4.8)

and

\[
P(Y_{i,j}^{\text{out}} < t_{i,j}|q_{i,j} = 1) = P(1 - \alpha < t_{i,j})p_c + P(Y_{i,j} < t_{i,j})(1 - p_c).
\]  

(4.9)

Note that here \( Y_{i,j} \) is not considered as a random variable but rather as the actual value of the image luminance at pixel \( (i,j) \). Using (4.8) and (4.9) and substituting in (4.1) we obtain the probability of binarization error

\[
P_{\text{err}} = p_c[p_0P(t_{i,j} < \alpha) - p_1P(t_{i,j} < 1 - \alpha)]
\]

\[
+ (1 - p_c)(p_0 - p_1)P(t_{i,j} < Y_{i,j}) + p_1,
\]  

(4.10)

as a function of the cumulative probabilities of \( t_{i,j} \). Substituting the actual expression of the cumulative distributions of \( t_{i,j} \) in (4.7) it is possible to write the probability of error as a function of \( F(y) \)

\[
P_{\text{err}} = p_c \sum_{k=0}^{n} w_k (p_0F(\alpha - t_k) - p_1F(1 - \alpha - t_k))
\]

\[
+ (1 - p_c) \sum_{k=0}^{n} w_k(p_0 - p_1)F(Y_{i,j} - t_k) + p_1,
\]  

(4.11)

where for compactness we introduced the binomial coefficients as \( w_k = \binom{n}{k}p^k(1-p_0)^{n-k} \). This probability of error depend on the distribution of the image, the QR code and the transformation parameters \( \alpha \) and \( p_c \). This dependence is reflected in the location of the discrete shifting parameters \( t_k \) which are a function of \( \alpha \) and \( n \) as well as in the binomial coefficient \( w_k \) that also depends on \( n = \lfloor p_c(64 \times 25) \rfloor \). Figure 4.1 (a) depicts the empirical probability of binarization error and the approximate expression in (4.1) for the image patch of Figure 4.1 (c). Despite the independence of the variables in (4.3), a good agreement between the model and empirical distributions for different values of \( \alpha \) and \( p_c \) is observed. One important observation about this distribution is that it presents a sharp transition for a particular value of \( \alpha \). Below this value, the probability of binarization error is mainly constant.
The transition value in the probability of binarization error can be estimated from the expected values of the threshold with respect to $\alpha$. If we impose the condition for the modified pixels that $\alpha < E[t_{i,j}]$ and $(1 - \alpha) > E[t_{i,j}]$ to recover the correct binary values at the modified pixels it is possible to obtain an inequality for the values of $\alpha$ in the minimum probability zone as a function of $p_c$. Since each threshold can be decomposed in two independent components $t_{i,j} = a_{i,j} + b_{i,j}$, its expected value can be found as

$$
E[t_{i,j}] = E[a_{i,j}] + E[b_{i,j}]
$$

$$
E[t_{i,j}] = p_c(1 + (1 - \alpha)p_c) + (1 - p_c)E[Y_{i,j}],
$$

(4.12)

imposing the conditions $\alpha < E[t_{i,j}]$ and $1 - \alpha > E[t_{i,j}]$ with the expected value of the threshold yield two inequalities on $\alpha$ given

$$
\alpha < \frac{1 - (p_c p_1 + E[Y_{i,j}](1 - p_c))}{(1 + p_c (p_0 - p_1))},
$$

(4.13)

$$
\alpha < \frac{p_c p_1 + E[Y_{i,j}](1 - p_c)}{(1 + p_c (p_0 - p_1))}.
$$

The maximum value of $\alpha$ in the low probability region is obtained as the minimum of these two bounds. This is not a deterministic bound because of the stochastic nature of $t_{i,j}$ but it holds with high probability as long as the threshold $t_{i,j}$ is tightly concentrated around its mean.

The region in the parameter space $(\alpha, p_c)$ for corresponding to the minimal probability of error is given by

$$
\alpha < \min(b_1(p_c), b_2(p_c)),
$$

(4.14)

where

$$
b_1(p_c) = \frac{1 - (p_c p_1 + E[Y_{i,j}](1 - p_c))}{(1 + p_c (p_0 - p_1))},
$$

(4.15)

and

$$
b_2(p_c) = \frac{p_c p_1 + E[Y_{i,j}](1 - p_c)}{(1 + p_c (p_0 - p_1))},
$$

(4.16)

Figure 4.1(b) shows the minimal probability region in the parameter space with the cross corresponding to the combination used in Figure 4.1(c).
Figure 4.1: a) Probability of binarization error for different values of $p_c$. The vertical lines correspond to the transition threshold of (4.14). b) The shaded region correspond to the region below the bound (4.14). c) Color patch transformed with the pair of parameter marked in b), the (upper left) Figure is the original image, (upper right) is the QR code, (lower left) is the embedded code and (lower right) the binarization of this code.

A. Optimal Transformation Parameters

The optimal transformation parameters $\alpha$ and $p_c$ should be selected to minimize the probability in (4.1) and also to maximize the visual fidelity of the embedding. The appropriate way to address this is through the formulation of an optimization problem with a cost function that penalizes the probability of binarization error as well as the visual distortion of the embedding. The function proposed is formed by the combination of the probability of error derived in (4.1) and the mean square error between the original gray scale image and the embedding image $\| Y^{out} - Y \|_F^2$. Based on the previous analysis we can enforce the constraint such that all feasible solutions should belong to the minimal probability region of (4.15). The optimization problem considering all this aspects is given by

$$
(\alpha^*, p_c^*) = \arg \min_{\alpha, p_c} \ P_{err}(\alpha, p_c) + \frac{\lambda}{2} \ \| Y^{out} - Y \|_F^2 \\
\text{subject to} \ \alpha < \min(b_1(p_c), b_2(p_c))
$$

(4.17)

where the regularization parameter $\lambda$ controls the trade off between decoding robustness and visual fidelity. This cost function is nonlinear and can have many local minimums, however it is possible to obtain acceptable solutions by solving the problem
for different initial conditions and choosing the solution with minimum cost. In order to guarantee decodability a constraint $P_{err} < P_{\text{min}}$ in the probability of error can be added, where the value of $P_{\text{min}}$ should depend on the robustness of the error correction mechanism in the QR code. It is important to note that the optimal values found through the optimization are local and only apply to the corresponding block of patches. The method used to smoothly extend their values to close neighborhoods is the bicubic interpolation.
Chapter 5

IMAGE EMBEDDING RESULTS

This chapter presents the result of processing the images with the optimal threshold that are the results of the optimization in \((4.17)\). The proposed algorithm appropriately selects the color of each pixel, such that the dark and light regions are separable even though each individual pixel might have values outside the decodable thresholds. The algorithm fixes this problem by assuring that the majority of the pixels in black regions have dark while for the white regions have light tone. This effect commonly used in halftoning of digital images can be also used here constraining the algorithm to generate images that are correctly decoded while being visually pleasant. Examples of the embedded images are depicted in Figure 5.1, while Figures 5.2, 5.3, 5.4 depicts the results obtained when \(p_c\) and \(\lambda\) parameters are change.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{qr_codes.png}
\caption{(Top row)Four QR codes generated by our method. (Bottom row) Original Images.}
\end{figure}
**Figure 5.2:** (Top row) Variation of the $p_c$ parameter to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for $\alpha^*$. 
**Figure 5.3:** (Top row) Variation of the $\lambda$ parameter to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for $\alpha^*$. 
Figure 5.4: (Top row) Variation of $p_c$ and $\lambda$ parameters to visualize the Influence in the quality of resulting image. (Bottom row) Distribution of the optimal values for $\alpha^*$. 
Chapter 6

CONCLUSIONS

The proposed method takes as input an image or logo and a QR code, and embeds the logo in the QR code in such a way that the decoder correctly detects dark and light regions. We render different scales of gray by the use of digital halftoning techniques perfectly suited for printing in physical media as well as for rendering in a digital screen. The main difference of this method with respect to already available solutions is the capability of covering the whole area of the QR code with the logo or image. There are some approaches in which less pixels that those allowed by the correcting capabilities of the code are modified, but these lead in general to a small logo in the center of the code not using the full area available. The output halftone of the system, is guaranteed to be decodable. The robustness of the decodability is assured by tuning the set of parameters $N_c$, $\alpha$ in the algorithm allowing to go from the original logo to the black and white QR code. The future work includes to maintain fixed the values of the center of the QR code, while the other parts are modified based on the halftone mask. This method will permit to distribute more uniformly the pixels of the images resulting in a better quality of the output image.
BIBLIOGRAPHY


Appendix

DISCLAIMER

The figures presented here, where not endorsed by the trademark owners and are used here as fair use to illustrate the quality of the achieve improvement of the aesthetic appearance of QR codes. Coca-Cola is a trademark of the Coca-Cola Company, which does not sponsor, authorize or endorse the images in this thesis.

©2013 The Coca-Cola Company. All rights reserved.