QR Images: Optimized Image Embedding in QR Codes

Gonzalo J. Garateguy, *Student Member, IEEE*, Gonzalo R. Arce, *Fellow, IEEE*, Daniel L. Lau, *Senior Member, IEEE*, and Ofelia P. Villarreal, *Member, IEEE*

***Abstract*— This paper introduces the concept of QR images, an automatic method to embed QR codes into color images with bounded probability of detection error. These embeddings are compatible with standard decoding applications and can be applied to any color image with full area coverage. The QR information bits are encoded into the luminance values of the image, taking advantage of the immunity of QR readers against local luminance disturbances. To mitigate the visual distortion of the QR image, the algorithm utilizes halftoning masks for the selection of modified pixels and nonlinear programming techniques to locally optimize luminance levels. A tractable model for the probability of error is developed and models of the human visual system are considered in the quality metric used to optimize the luminance levels of the QR image. To minimize the processing time, the optimization techniques proposed to consider the mechanics of a common binarization method and are designed to be amenable for parallel implementations. Experimental results show the graceful degradation of the decoding rate and the perceptual quality as a function the embedding parameters. A visual comparison between the proposed and existing methods is presented.**

***Index Terms*— QR codes, image embedding, halftoning.**

I. Introduction

QUICK response (QR) codes [1], [2] have rapidly emerged as a widely used inventory tracking and identification method in transport, manufacturing, and retail industries [3]. Their popularity is due to the proliferation of smart phones, capable of decoding and accessing on line resources as well as its high storage capacity and speed of decoding. QR codes are used in a variety of applications, such as accessing websites, download personal card information, post information to social networks, initiate phone calls, reproduce videos or open text documents. This versatility makes them a valuable tool in any industry that seeks to engage mobile users from printed materials. Not surprisingly QR codes have been widely adopted in the marketing and publicity industry thanks to the advantage they provide in tracking the performance of publicity campaigns.

An important problem of QR codes is its impact on the aesthetics of publicity designs. The square shapes and limited color tolerance, severely impairs their integration into billboard designs or printed materials. This challenge has generated great interest for algorithms capable of embedding QR codes into images without loosing decoding robustness. There have been several efforts to improve the appearance of such embeddings [4]–[15] which can be classified in two categories, methods that modify the luminance or color of image pixels and methods that replace QR modules.

The methods presented in [6] and [7] base the strategy on finding the best group of QR modules to substitute by the image or logo in the QR code. Other methods take advantage of unused modules in the padding regions [8]–[10] to introduce the image without affecting the decoding robustness. In [7] the impact of replacing code modules with image pixels was studied. The authors concluded that to retain high rate of decodability, the ratio between image and code area should be approximately proportional to the correction capacity of the code. It was also found that superimposing images over finder or alignment patterns severely decrease the probability of correct decoding. As a consequence it is common for logos or images to be located at the center of the code for these methods as depicted in Fig. 1(a). In general these approaches do not take advantage of the codeword generation process and this imposes restrictions in the location of modified modules. This problem was addressed by recently developed techniques [4], [11], [12] which manipulate the Reed Solomon encoding procedure to maximize the area coverage without reducing the correction capacity.

The second category of embedding algorithms is based on the modification of the pixel's luminance. The approach in [5] chooses central pixels of each module to modify its luminance since this is the area usually sampled by the decoder. This approach provides an adequate trade off between robustness and visual distortion but the number of center pixels modified is in general a large proportion of the module area and creates undesirable low pass artifacts [see Fig. 1(c)]. A method which implements a similar idea is LogoQ [13]. In [16] a method to introduce colors into QR codes was proposed with the goal of increasing its data capacity and a detailed study of the interference between different color layers was presented. Two embedding methods which are related to the method presented here are [14] and [15]. In [14] the author chooses the modified pixels in the code based on the average luminance of

Manuscript received September 6, 2013; revised December 31, 2013 and April 14, 2014; accepted April 16, 2014. Date of publication May 2, 2014; date of current version May 20, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Karsten Mueller.

G. J. Garateguy is with Mathworks, Natick, MA 01760 USA (e-mail: ggarateg@udel.edu).

G. R. Arce and O. P. Villarreal are with the Department of Electrical and Computer Engineering, University of Delaware, Newark, DE 19711 USA (e-mail: ggarateg@udel.edu; ofeliap.villarreal@gmail.com).

D. L. Lau is with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506 USA (e-mail: dllau@engr.uky.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2321501

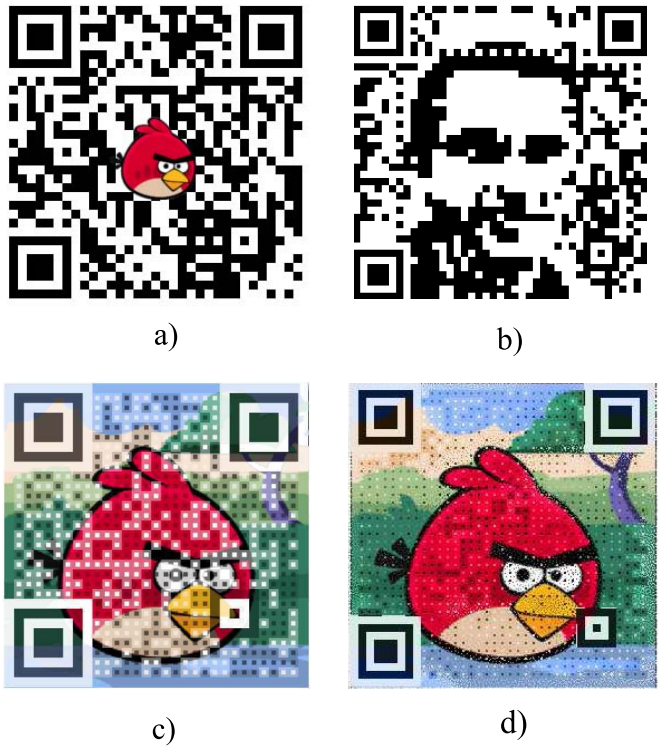


Fig. 1. (a) Substitutions of QR modules by a logo, (b) QArt code designed using the method in [4], (c) luminance modification algorithm [5], $J = 0.4149$ (d) proposed algorithm, $J = 0.3989$. The distortion metric J is defined in (16).

image pixels and two user defined thresholds that regulate the reliability of the embedding. QR blocks are still visible but the visual impact is reduced with respect to [5]. The method presented in [15] generates a binary embedding by subdividing each QR module in a set of 3×3 pixels and setting the central pixel to the value of the QR code while the remaining pixels are selected to minimize the probability of error and generate a halftone of the image. In this case the probability of error is evaluated empirically using a large database of random embeddings.

The main challenge of any embedding method is the fact that they should be decodable by standard applications. The embedding of image pixels introduce changes in the luminance of the code, distorting the binarization thresholds and thus increasing the probability of detection error. The second challenge concerns the problem of using the entire area of the code in which the image or logo is to be embedded. This cannot be achieved by simply replacing information modules with the desired image since the number of modules that can be replaced is at most proportional to the correction capacity of the code [7]. A good embedding method, should minimize the number of corrupted modules and use the greatest possible area while keeping visual fidelity to the original image.

This paper aims at the above cited goals by introducing QR images, an optimization based approach for embedding color images into QR codes. The algorithm proposed is based on the selection of a set of modified pixels using a halftoning mask. The concentration of pixels and its corresponding luminance are optimized to minimize a visual

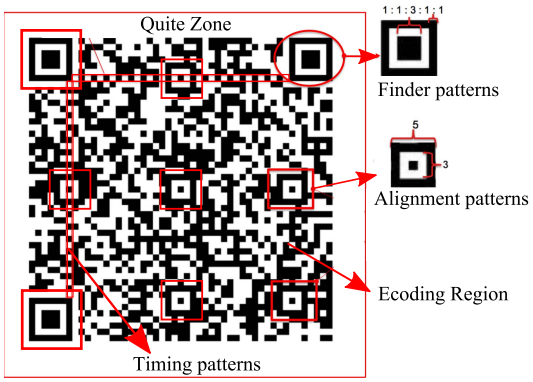


Fig. 2. QR code regions with the location of finder and alignment patterns highlighted in red.

distortion metric subject to a constraint in the probability of error. This algorithm can be applied to any color image and QR code with full area coverage and bounded probability of error. A novel contribution of this paper is the use of halftone masks to distribute the modified pixels and the introduction of a probabilistic model to predict the distortion generated by the embedded image. This allows to control the trade off between image quality and decoding robustness by setting a few parameters such as the number of modified pixels at the center of each QR module and the maximum allowed probability of error. Fig. 1(d) depicts the results of embedding a QR code using the method proposed in this paper which is visually more pleasant than the embedding in Fig. 1(d) and presents a lower visual distortion according to the metric defined in (16).

Sections II and III presents the structure and decoding procedures for QR codes since they are central to understand the possible modification strategies. Section IV presents the concept of QR images and the luminance manipulation algorithm and Section V describes the probability of error models as a function of the embedding parameters. Finally Section VI and Section VIII present the optimization algorithm and the embedding results of applying the algorithm to several images and logos.

II. Characteristics of QR Codes

The patterns and structures inside a QR code have well defined functions which include symbol alignment, sampling grid determination, and error correction. The information is encoded in square black and white modules of several pixels wide. Finder patterns play a central role in the speed and success of decoding and are located in three corners of the symbol. Fig. 2 shows the three main regions in the QR symbol structure: function pattern region, encoding region and the quiet zone which is a guard region located on the outside of the symbol.

1) Function Pattern Region: Finder and alignment structures are essential to locate, rotate and align the QR code. The former ones are designed to have the same ratio of black and white pixels when intersected by a line at any angle, allowing to easily detect rotated or inverted codes. Alignment patterns are used to determine the sampling grids from which

codewords are extracted and they are easily identifiable as concentric square structures evenly distributed along the code area.

2) Encoding Region: The code area delimited by finder patterns is denoted as the encoding region, where data, parity modules and decoding information is stored. This area is divided into codewords consisting of blocks of 8 QR modules. Two dimensional shapes of these codewords depend on the version of the code and are designed to optimize area coverage.

3) Data Capacity and Error Correction: Different types of QR codes defined in the standard [1], are identified by their version and error correction level. The version of the QR code determines its size and goes from 21×21 modules for version 1 up to 177×177 for version 40. QR codes use Reed Solom code for error correction and there are 4 types of error correction L, M, Q and H that allow to correct up to 7%, 15%, 20% and 30% of codewords in error respectively.

III. Decoding Algorithm

After acquiring the image and calculating its luminance from the RGB components, the decoding process continues with three basic stages: binarization, detection, and decoding of the bit stream. In the binarization stage, the gray scale image captured by the camera is segmented into black and white pixels. This binary image is used to determine the QR modules centers and the sampling grid from which the codewords are extracted. After this process, detected codewords are corrected using the Reed Solomon algorithm and then decoded according to its format and version.

A. Threshold Calculation for Binarization

A salient feature of QR codes which plays a central role in their decoding speed is the use of binarization as the first step in the decoding process. Binary images are obtained by thresholding the gray scale image as

$$I_B[i,j] = \begin{cases} 1 & \text{if } Y_{i,j} > t_{i,j} \\ 0 & \text{if } Y_{i,j} \leq t_{i,j} \end{cases} \quad (1)$$

where Y is the image captured by the camera, $t_{i,j}$ is the threshold assigned to pixel $[i,j]$ and $I_B[i,j]$ is the binary output. Determining the values of the thresholds $t_{i,j}$ for an optimal binarization is a challenging task due to the variability in lighting conditions. The QR code standard does not define specific binarization methods and these are usually selected making a trade off between speed and quality. Thresholds can be calculated globally using all the pixels in the image or locally considering only reduced windows. Global segmentation methods as [17], [18] have been tried and proved ineffective since they require high level of contrast in the acquired image. Local thresholding strategies on the other hand are better suited to illumination variations and have been successfully applied to QR code binarization [19], [20]. Adaptive methods such as the one presented in [21] show better binarization accuracy but at the cost of increased computational complexity which is not always possible to satisfy in embedded devices.

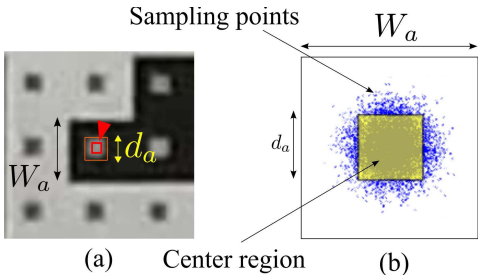


Fig. 3. (a) Image of a QR module showing the length of the module and the center region in pixels. (b) Diagram of a module showing possible sampling points.

B. Mean Block Binarization Method

One of the most popular libraries for QR code generation and reading is the open source Zxing library [22]. The thresholds used in the binarization functions of this library, are calculated through a hybrid method that use the average luminance in a set of overlapping square windows. The captured image is divided into non-overlapping blocks $B_{m,n}$ of 8×8 pixels and then the average luminance in overlapping sub windows of 5×5 blocks is calculated according to

$$T_{m,n} = \frac{1}{25\times64} \sum_{p=m-2}^{p=m+2} \sum_{q=n-2}^{q=n+2} \sum_{(k,l)\in B_{p,q}} Y[k,l]. \quad (2)$$

The averages calculated for each block $B_{m,n}$ are assigned to the pixels in the block as $t_{i,j} = T_{m,n}$ for $[i,j] \in B_{m,n}$. Segmentation performance is inferior to the method presented in [19] but this method has the advantage of being computationally less expensive. This is the technique assumed in the following sections to develop the probability of error model and the QR embedding algorithm.

C. Sampling Grid and Probability of Sampling Error

Once the binary image I_B is obtained, codewords are extracted by sampling on a grid estimated using finder and alignment patterns. The points in this grid are generated by drawing parallel lines between the estimated centers of finder and alignment patterns and the spacing between lines is set to the estimated width of a QR module W_a. For larger code sizes, multiple sampling grids are used to compensate for local geometric distortions.

In order to safely detect the binary value, the luminance around the center of the module should be clearly defined. If we define a region of size $d_a \times d_a$ pixels centered in the QR module, the probability of sampling outside this region can be obtained by assuming a Gaussian distribution of the sampling point around the center and integrating outside the region as depicted in Fig. 3. In the following section we model the sampling distribution as a Gaussian with $\sigma = W_a/4$. The probability of sampling error denoted by p_s can be precomputed for different sizes of W_a and d_a to be used in the embedding algorithm.

IV. Aesthetic Enhancement of QR Codes

The QR code embedding technique introduced here, encodes the QR code value on the luminance values of the

image in such a way that the average luminance is increased for light regions in the code and decreased for dark regions. In fact any embedding algorithm tailored for a standard decoder must be a variation of this type since the binarization thresholds are usually calculated as local averages of pixels luminance. The parameters of the modification consist of selecting the location of modified pixel and its luminance or to keep the QR module pixels unchanged. In this paper we consider the last possibility as a particular case in which the concentration of modified pixels in a region is close to 1. The embedding method proposed consists of two components. The first is the use of halftoning techniques for the selection of modified pixels allowing to break and reduce the coarse square structures of the code. The second component is the modification of the luminance levels to minimize the image distortion. This modification parameters are optimized independently in local windows as described in Section VI in order to accelerate the embedding and leverage the correlation between the luminance of the image and the code.

A. Halftoning Techniques

The method proposed to select modified pixels is based on halftoning techniques in order to minimize the appearance of blocks while preserving high frequency details. If modified pixels are randomly but uniformly distributed in space, the visual impact of the embedding is minimized since these patterns concentrate most of their energy at higher frequencies where the human visual system is less sensitive. This effect is commonly used in digital halftoning [23] where different algorithms to generate even distributions of points with particular spectral properties have been proposed. Examples of such algorithms are, error diffusion [24], [25], blue noise masks [26]–[28], green noise masks [26], or direct binary search [29]. Error diffusion has a long history in the printing industry where it has been used for the last 30 years; however, being a recursive algorithm, the processing time increase considerably for very large images. Blue and green noise masking techniques in contrast, generate high frequency binary patterns by thresholding a carefully designed multilevel array. These patterns can be computed in parallel which greatly increase the speed of generation. Many algorithms to design halftoning masks were presented in the literature such as, Void and Cluster [30], direct binary search [29], green noise masks [26], [31], blue noise multitone dithering [32] and techniques based on centroidal voronoi tessellations [28]. Each of these techniques have different computational complexities, but since the mask design process is performed offline the speed of pattern generation does not change.

B. Pixel Selection

Ideally, only the pixels at the center of the QR module are relevant for a correct decoding, however, due to errors in the determination of the sampling grid, adjacent pixels play an important role in the decoding process. To account for this we make a distinction between the pixels in the QR module. A square of size $d_a \times d_a$ is always selected for modification and the remaining pixels in the module are selected using a halftone mask. The size of the square of center pixels is selected by the user and regulates the robustness of the embedding but also affects its visual quality. The distribution of modified pixels in non central pixels is generated by thresholding a blue or green noise mask to generate a binary pattern with a concentration of dots of p_c. To simplify the notation we denote by I_{p_c} the binary pattern generated by the halftoning mask and by M a mask that is 1 on the set of central pixels and 0 otherwise.

C. Luminance Modification

After selecting the pixels, its luminance is modified to one of four possible levels $\alpha, \beta, \alpha_c, \beta_c$. The luminance of the embedded image $Y^{out}_{i,j}$ at (i, j) is selected as a function of the QR code value $q_{i,j}$ and the luminance of the original image $Y_{i,j}$ as described in equation (3). This transformation changes the luminance of the pixels that are selected according to the halftone distribution and keep the remaining pixels in the image unchanged. The pixels at the center of the QR module are assigned different luminance levels, since they play a central role in the detection of binary values when the sampling accuracy is high.

$$Y^{out}_{i,j} = \begin{cases} \beta & \text{if } M_{i,j} = 0,\ q_{i,j} = 1,\ I_{p_c,(i,j)} = 1 \\ \alpha & \text{if } M_{i,j} = 0,\ q_{i,j} = 0,\ I_{p_c,(i,j)} = 1 \\ \beta_c & \text{if } M_{i,j} = 1,\ q_{i,j} = 1 \\ \alpha_c & \text{if } M_{i,j} = 1,\ q_{i,j} = 0 \\ Y_{i,j} & \text{otherwise.} \end{cases} \tag{3}$$

D. Color Optimization

Since the goal is to embed the codes into color images, we need to establish a rule to select the corresponding color vector for each modified luminance since there is an associated subspace of colors in the RGB space for each luminance value. To determine the best possible color that fulfills the luminance constraint, it is necessary to measure the color differences into a perceptually uniform color space [33]. The HSL color space is selected because it involves simpler computations than other color spaces while it is still capable of representing perceptual difference using euclidean metrics. To obtain the RGB values of a modified pixel given the luminance target $l_t \in [0, 1]$, the original color is transformed into the HSL color space and then the L component is optimized keeping S and H fixed until reaching the desired luminance $Y = l_t$. The relationship between the luminance defined as $Y = 0.2989R + 0.5870G + 0.114B$ and the

$$L = \frac{\min(R, G, B) + \max(R, G, B)}{2}$$

component in the HSL color space is a piecewise, linear and monotone function $Y = f(L)$. If the weight luminance vector is defined as $w = [0.298, 0.587, 0.1140]^T$, then $f(L)$ is given by

$$Y = f(L) = w^T T^{-1}(H, S, L), \tag{4}$$

where $(R, G, B) = T^{-1}(H, S, L)$ is the backward transformation from HSL to RGB color spaces. The optimal value

Algorithm 1 Color Modification for Luminance Target

Require: (R, G, B) pixel and target value of $Y^{out} = l_t$

$(H, S, L) \leftarrow T(R, G, B)$

find $L^* = \underset{L}{\text{argmin}} |f(L) - l_t|$

$(R^*, G^*, B^*) \leftarrow T^{-1}(H, S, L^*)$

return (R^*, G^*, B^*)

of L is obtained from the luminance target $Y^{out} = l_t$ as the solution of

$$L^* = \underset{L}{\text{argmin}} |f(L) - l_t|. \quad (5)$$

Once the optimal value L^* is determined, the new RGB components of the pixels are obtained by using the forward transformation between the HSL and RGB color spaces $(R^*, G^*, B^*) = T^{-1}(H, S, L^*)$. Summarizing the different steps, the transformations to obtain the RGB value of a modified pixel, given the target luminance value l_t is described in algorithm 1.

V. Probability of Error Models

Pixels of the embedded code are a combination of pixels from the image and the QR code as described in (3). This mixture distorts the binarization thresholds with respect to the black and white QR code, and in general increases the probability of binarization error. The problem can be minimized by optimizing the luminance parameters, but this presents a trade off between decoding robustness and image quality. Efficient optimization methods require a tractable model for the probability of error defined in local neighbourhoods to leverage the parallel nature of the problem. The probability model presented here is based on reasonable assumptions and yields an expression for the probability of error that is smooth and continuous and can be easily integrated into different optimization algorithms. This model considers two regimes, which correspond to the precise sampling of QR modules centers and the case when all the pixels in the module are indistinguishable. Two independent probability models are constructed and then combined to yield a general model.

A. Probability of Binarization Error Model

The probability of binarization error is defined as the probability of sampling the incorrect binary value at any pixel in the QR module. This probability is influenced by different factors such as the local distribution of luminance values in the image, the distribution of pixels in the QR code and the parameters of the luminance transformation. To formulate a model of the probability of error, consider the detected binary value at pixel (i, j). The probability of error at this location is given by

$$\begin{aligned} P_{Berr} = {} & P(\text{decide } q_{i,j} = 1 | q_{i,j} = 0) p_0 \\ & + P(\text{decide } q_{i,j} = 0 | q_{i,j} = 1) p_1. \end{aligned} \quad (6)$$

where $p_0 = P(q_{i,j} = 0)$ and $p_1 = P(q_{i,j} = 1)$ are the probabilities of having a black or white pixel in the code respectively. The detector makes a binary decision on the value of the code $q_{i,j}$ based on the threshold $t_{i,j}$ and the embedding luminance $Y^{out}_{i,j}$. Taking this into account, the probabilities of miss detection in (6) can be expressed as

$$\begin{aligned} P(\text{decide } q_{i,j} = 1 | q_{i,j} = 0) &= P(Y^{out}_{i,j} > t_{i,j} | q_{i,j} = 0) \\ P(\text{decide } q_{i,j} = 0 | q_{i,j} = 1) &= P(Y^{out}_{i,j} < t_{i,j} | q_{i,j} = 1). \end{aligned} \quad (7)$$

The gray scale image Y is modeled as a bi-dimensional random field, and the thresholds as correlated random variables. The exact calculation of the probabilities in (7) involves the knowledge of the joint distribution of the image and threshold values which in practice can be quite challenging to calculate. Instead of using this general relation a simplified model is proposed here, assuming independence of the components of the embedding luminance $Y^{out}_{i,j}$. The value of $Y^{out}_{i,j}$ at location (i, j) can be decomposed as the combination of several random variables that are related to the luminance parameters of the transformation. In this model we don't differentiate between central and non-central pixels and consider that the contribution of central pixels to the threshold is minimal.

The expression for the output image embedding under these assumptions is given by

$$Y^{out}_{i,j} = [\beta q_{i,j} + \alpha(1 - q_{i,j})] I_{p_c,(i,j)} + Y_{i,j}(1 - I_{p_c,(i,j)}). \quad (8)$$

The first component of this expression is the binary random variable $I_{p_c,(i,j)}$ that selects the pixels being modified and models the stochastic nature of the blue binary pattern. Despite the fact that the location of modified pixels are correlated by the binary pattern constraints, it is reasonable to model them as independent Bernoulli random variables given that a sufficiently large window is considered. Furthermore, independence is also assumed between QR code values, $q_{i,j}$ and luminance values $Y_{i,j}$ in the original image. Finally the variables $Y_{i,j}$ are modeled as independent random variables following a local distribution based on the size and location of the window in the image. Combining these assumptions and the expression for the binarization thresholds, it is possible to derive (see Appendix A) a model for the probability distribution of the thresholds given by

$$P(t_{i,j} = t) = \sum_{k=0}^{n} f(t - t_k) \binom{n}{k} p_1^k p_0^{n-k}. \quad (9)$$

where $n = \lfloor p_c(64 \times 25) \rfloor$ is the total number of modified pixels in the window of blocks defined in (2),

$$t_k = \left[\frac{k\beta}{25 \times 64} + \frac{(n-k)\alpha}{25 \times 64} \right]$$

are the possible outcomes from the sum of n modified luminance pixels $\beta q_{i,j} + \alpha(1 - q_{i,j})$ and f is the convolution of $f_Y(x)$ corresponding to the mean value of unmodified pixels and $f_n(x)$ corresponding to the distribution of the Gaussian noise at the detector. This model is central for the tractability of the decision probabilities in (7) and yields a smooth and

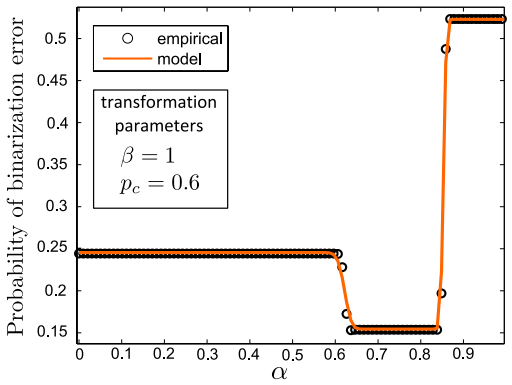


Fig. 4. Probability of binarization error (PBE) for a randomly selected window as a function of α for fixed β and p_c. The variance of the noise used was $\sigma_\eta = 0.1$.

continuous expression for the probability of binarization error given by

$$P_{Berr} = p_c \sum_{k=0}^{n} w_k \left(p_0 F(\alpha - t_k) - p_1 F(\beta - t_k)\right) + (1 - p_c) \sum_{k=0}^{n} w_k (p_0 - p_1) F(Y_{i,j} - t_k) + p_1 \quad (10)$$

where the coefficients

$$w_k = \binom{n}{k} p_1^k p_0^{n-k}$$

are introduced for compactness. Fig. 4 depicts the agreement between the probability model and the empirical probability of binarization error calculated by adding Gaussian noise with the appropriate variance to the embedding and then calculating the probability of error by counting the number of discrepancies between the QR code and the binarized embedding.

B. Probability of Detection Error

If we assume that the pixels at the center of each module can be sampled accurately, then the probability of error is given by

$$P_{Derr} = P(\text{decide } q_c = 1 | q_c = 0) p_0 + P(\text{decide } q_c = 0 | q_c = 1) p_1 \quad (11)$$

where q_c is the value of the QR code module at its center and $p_0 = P(q_c = 0)$, $p_1 = P(q_c = 1)$ are the probabilities of having the corresponding QR module inside the local window. There is a slight difference with respect to the expression in (6) since here we consider the probability of detecting the incorrect value but only at the center position of the module. This is a much relaxed requirement than the one in (6) since only the mean value of non-central pixels is important. This naturally leads to a different probability model where the miss detection probabilities in (11) reduce to

$$P(\text{decide } q_c = 1 | q_c = 0) = P(\alpha_c > t)$$
$$P(\text{decide } q_c = 0 | q_c = 1) = P(\beta_c < t). \quad (12)$$

Considering the luminance transformation in (3) and using the four luminance levels. The binarization thresholds at the detector are given by $t = \mu + b + \eta$ where

$$\mu = \frac{\alpha n_\alpha + \beta n_\beta + \alpha_c n_{\alpha_c} + \beta_c n_{\beta_c}}{N}$$

and b is the mean value of unmodified pixels in the window. The number of pixels in each modified luminance level is denoted by, $n_\alpha, n_\beta, n_{\alpha_c}, n_{\beta_c}$ and N is the total number of pixels. By the considerations made in Section IV-C the variable b can be regarded as a Gaussian random variable with mean $\mu_b = E[Y_{1,j}](1 - p_c)$ and variance

$$\sigma_b^2 = \frac{Var[Y_{i,j}]}{N}(1 - p_c)$$

independent of the noise η. Therefore the probability distribution of t is also Gaussian with mean $\mu_t = \mu + \mu_b$ and variance $\sigma_t^2 = \sigma_\eta^2 + \sigma_b^2$ and its cdf can be expressed as a function of the Gaussian normal cdf,

$$P(t < x) = \Phi\left(\frac{x - \mu_t}{\sigma_t}\right).$$

Finding the probability of detection error then reduces to combining this result with the expressions in (7) and (11) yielding

$$P_{Derr} = \Phi\left(\frac{\alpha_c - \mu_t}{\sigma_t}\right) p_0 - \Phi\left(\frac{\beta_c - \mu_t}{\sigma_t}\right) p_1 + p_1. \quad (13)$$

C. Global Probability of Error

The previous probability models define two clear regimes in the detection of binary values, however, there are cases in in which only a faction of the sampled pixels lie inside the center regions with modified luminance α_c or β_c. However, since the probability of sampling error can be precomputed for a given module (W_a) and center size d_a, it is possible to find the global probability of error by conditioning on the event of successful sampling of the center region. The global probability of error is then given by

$$P_{err} = P_{Berr} p_s + P_{Derr}(1 - p_s). \quad (14)$$

VI. Quality Metrics

Based on the probability models developed, is possible to accurately determine the probability of error for a given set of luminance parameters and guarantee the decodability of the embedding. However, it is necessary to define appropriate metrics to quantify also its visual quality. In spite of the complexities of the human visual system (HVS) a largely used quality metric to measure image similarity is the mean squared error (MSE) since it has many desirable mathematical properties such as convexity and differentiability, however it does not always correlates with visual quality [34]. There have been efforts to overcome the limitations of the MSE such as the SSIM index [35] which explicitly considers different luminance, contrast and structure measures. For the case of halftones the most commonly used metric is based on the MSE between two filtered versions of the original image and the halftone where the filter usually models the low pass

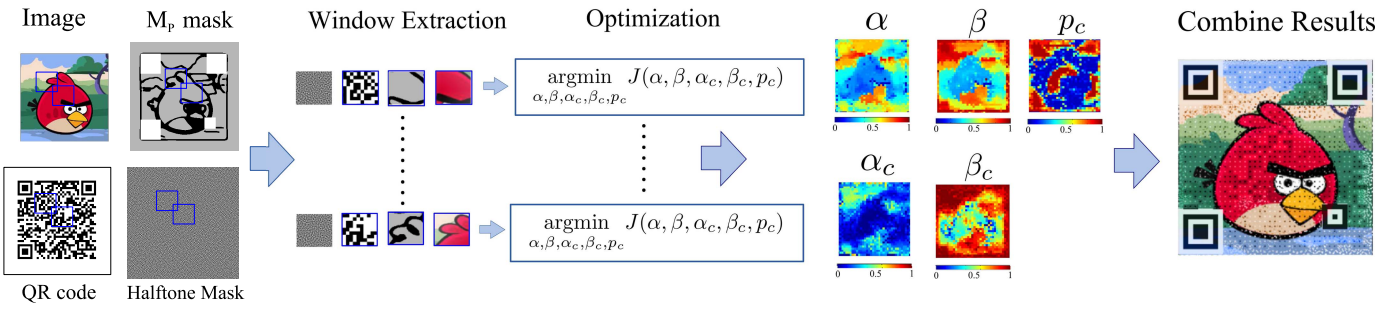


Fig. 5. Different stages of the QR embedding procedure. The inputs are the original image, the QR code, the halftone mask, the masks used for image prioritization and global value of P_{max} and d_a. These images are divided in local windows and then optimized independently and in parallel. Finally the results are interpolated to combine them in the final result.

characteristic of the HVS. One of the most used models for the HVS was proposed by Näsänen [36] and consist of an exponential model $H(f_r) = e^{-kf_r}$ where f_r is the radial frequency in cycles/degree. The corresponding discrete impulse response considering the viewing distance D in inches and the resolution R in dots/inch is given by

$$h[m,n] = \frac{k}{2\pi} \frac{1}{\left(\frac{k}{2\pi} + \left(\frac{180m}{\pi RD}\right)^2 + \left(\frac{180n}{\pi RD}\right)^2\right)^{3/2}} \quad (15)$$

where the constant k is determined by fitting empirical data to the model [36].

To evaluate the similarity between the image and the embedding, we propose a variation of the metric presented in [37] originally introduced to design structure aware halftones. The rationale for this selection is that blocks of the QR code should be less visible when viewed from a distance where the low pass characteristics dominate but at the same time the embedding should preserver important details when viewed at closer distances. This metric considers both the halftone visibility and image structure and allows to select the trade off between both by adjusting two weighting parameters. The general form of this metric is given by

$$J(Y^{out}, Y) = w_l G(Y^{out}, Y) + w_h(1 - MSSIM(Y^{out}, Y)). \quad (16)$$

where the term $G(Y^{out}, Y)$ gives more importance to the low pass error between the embedding and the image while the term $1 - MSSIM(Y^{out}, Y)$ measures the structure dissimilarity considering high frequency components and the overall structure of the image.

The weighting factors selected for all the simulations in this paper are $w_l = 1$ and $w_h = 1/2$ and its determination was based on several experiments with different images from which the best combination of parameters was selected. To make a faithful prediction of tone similarity the term G includes a model of the HVS and the difference between the images in calculated in the HSL color space since color similarity is an important component of the perceived similarity. The general expression for G is then given by

$$G(Y^{out}, Y) = \frac{1}{N} ||h * (L^{out} - L)||_F \quad (17)$$

where L^{out} and L are the lightness components of the embedding and original image in the HSL color space and N is the number of pixels in the window. Only the L component is considered here because S and H are unchanged in the color optimization. The filter h used in all the experiments is the one defined in (15) with parameters $D = 9$ inches and $R = 150$ dpi. Considering all the requirements, the final expression for the distortion metric used in the optimizations is then given by

$$J(\alpha, \beta, \alpha_c, \beta_c, p_c) = \frac{1}{N} ||h * (L^{out} - L)||_F + \frac{1}{2}(1 - MSSIM(Y^{out}, Y)). \quad (18)$$

VII. Parameter Optimization

In order to take advantage of local correlations between the luminance of the image and the values of the QR code, the optimization of the transformation parameters is performed independently using local overlapping windows. In our implementation we choose to set the size of the windows to 40×40 pixels centered around each image block of 8×8 pixels. This respects the proportions of the window defined in (2) and based on our experimental tests, yields a good trade off between robustness and quality for typical scanning distances.

The optimization of luminance parameters in local windows has other advantages besides leveraging the correlation between image and code luminance. For example it is possible to specify regions of the image which require higher visual quality or higher decoding robustness and set different probability bounds for each case.

A block diagram of the optimization process is depicted in Fig. 5. After the initial subdivision of the image, the code and the different masks, the 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 to match the size of the original image. This global map is then applied to the original image to obtain the QR embedding.

The local optimizations are formulated based on the quality metrics defined in (18) and the probability of error model in (14) as

$$\underset{\alpha,\beta,\alpha_c,\beta_c,p_c}{\text{argmin}} \; J_i(\alpha, \beta, \alpha_c, \beta_c, p_c) \quad \text{subject to } P_{err} < P_{max,i} \quad (19)$$

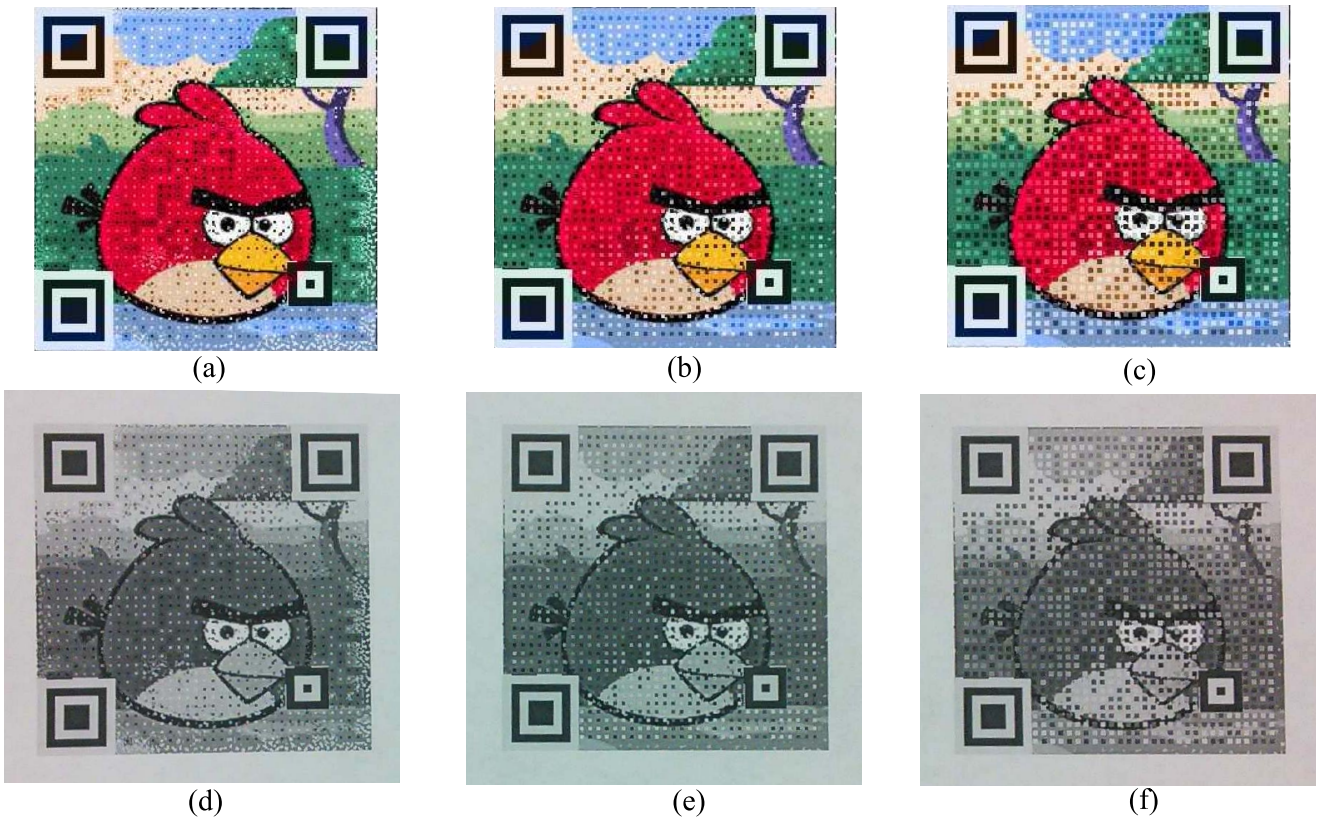


Fig. 6. (Top row) Image embeddings for (a) $d_a = 2$ pixels (J = 0.4), (b) $d_a = 3$ pixels (J = 0.4053), (c) $d_a = 4$ pixels (J = 0.4194). The QR modules in this example had a size of $W_a = 8$ pixels, and the global probability of error used was $P_{max} = 0.2$. A noise level of $\sigma_\eta = 0.1$ was assumed. (Bottom row) Images acquired using a Google Nexus 7 tablet at 12in from the target. The images were printed in a resolution of $150dpi$.

The input parameters to the optimizations are the global parameter P_{max} which determines the maximum admissible probability of error and the size of the center region in pixels, d_a. The parameter $P_{max,i}$ in the local optimization (19) is determined as a function of the local features of the image. In particular here, we define a mask M_P which has value -2 in regions where the image presents an edge, 1 in the function pattern regions and 0 elsewhere. The value of $P_{max,i}$ is calculated as

$$P_{max,i} = (1 - \text{mean}(M_{P,i}))P_{max} \tag{20}$$

where $M_{P,i}$ is the corresponding patch of M_P in the local window.

The edges are detected using a canny edge detector and lowpass filtering to generate the edge regions. This is a common practice in other embedding algorithms such as [12] or [15]. The values of $P_{max,i}$ can be distributed based on a probability of error budget consistent with the user preferences and type of error correction provided by the QR code (*i.e.* L, M, Q or H).

A. Nonlinear Optimization Methods

The optimization problem in Eqn. (19) is solved by means of a logarithmic barrier method [38]. The solutions are found by solving the sequence of problems

$$\begin{aligned} &\underset{\alpha,\beta,\alpha_c,\beta_c,p_c,s}{\text{argmin}} \quad J(\alpha, \beta, \alpha_c, \beta_c, p_c) + \mu \log(s) \\ &\text{subject to } P_{err}(\alpha, p_c) - P_{max} + s = 0 \end{aligned} \tag{21}$$

for decreasing values of the regularization parameter μ. The advantage of this approach is that the starting points can be infeasible at the beginning and as the regularization parameter is decreased the sequence of solutions converges towards a feasible local minimum of the original problem in Eqn. (19). The solver used is based on a sequential quadratic programing with trust region method presented in [39] and implemented by the $fmincon$ function of MATLAB [40].

VIII. Image and Logo Embeddings

This Section presents the result of embedding different images into a QR code and the experimental decoding rates obtained with mobile devices in order to evaluate the effectiveness of the embedding method on a real setting.

A. Decoding Robustness

To determine the robustness of the embedding against non modeled disturbances such as blur and uneven illumination we conducted an scanning experiment. The experiment consisted of scanning the embeddings in Fig. 6 from a printed version in which each QR pixels was render at 150 dpi and also from a laptop screen with a smart phone and a tablet. The embeddings generated had different sizes of the central region corresponding to $d_a = 2$, $d_a = 3$ and $d_a = 4$ pixels with a corresponding module size of $W_a = 8$ pixels and the halftone mask used was a green noise mask [41] to minimize the dot gain effect introduced by the printer. The devices used to scan

Fig. 7. Comparison of the embedding of different images and logos into a QR code. (a) Embedding of an image for a center region size of $d_a = 2$ pixels and a global probability of error of $P_{max} = 0.2$ using a blue noise mask (b) Embedding of a logo for $d_a = 1$ and $P_{max} = 0.15$ using a clustered dot mask aligned with the centers of the QR code modules. (c) Embedding of a logo for $d_a = 3$ and $P_{max} = 0.15$ using a blue noise mask. The embedding in (d) and (e) where generated using the method in [5] while the embedding in (f) corresponds to [14].

the codes were a smart phone Samsung Galaxy S I897 and a Google Nexus 7 tablet and the app used was Barcode Scanner[1] which is based on the Zxing Library. The recognition times where in general in the order of 5 to 10 seconds for the embeddings in Fig. 6(a) and Fig. 6(b) and in the order of 1 to 2 seconds for the embedding of Fig. 6(c) comparable to the method in [5]. The effective scanning distance depends on the resolution of the device, and in general is necessary to have multiple image pixels per QR module pixel for a successful decoding. This is in line with the assumption that QR pixels can be individually resolved, used to develop the probability models. Examples of the images captured by the tablet are depicted in Figs. 6(d), 6(e), and 6(f). These pictures were taken at an average distance of 9 inches from the target with approximately 10 pixels per QR module. The embeddings with smaller center regions where more susceptible to error and the scanning distance limited to 1 or 2 inches around the distance in which a QR module pixel is represented by a pixel in the captured image. For images with larger central regions such as 6(c). The range of successful scanning is extended since the assumption about correct sampling of the centers applies to a broader range of conditions. One possible explanation for the reduced robustness of the embedding is 6(a) and Fig. 6(b) is that the noise level used to design the embedding underestimates the contributions of poor illumination or camera blur.

[1]https://play.google.com/store/apps/details?id=com.google.zxing.client.android

TABLE I

THE MSE AND MSSIM ARE CALCULATED OVER THE LUMINANCE OF THE EMBEDDING AN THE ORIGINAL IMAGE. THE MSE PRESENTED HERE IS NORMALIZED BY THE NUMBER OF PIXELS IN THE IMAGE, $MSE = \frac{1}{\sqrt{N}}||(Y^{out} - Y)||_F$

Fig 7	(a)	(d)	(b)	(e)	(c)	(f)
J (18)	**0.3446**	0.4171	**0.3794**	0.3950	**0.4508**	0.9091
MSSIM	**0.5505**	0.3716	**0.5590**	0.4793	**0.3792**	0.0802
MSE	**0.1954**	0.2254	**0.2892**	0.3091	**0.2505**	0.6059

The values of P_{max} selected should be smaller than the corresponding decoding robustness of the code in order to have a successful decoding. In the case of Fig. 6, the code used had a correction level H (30% correction capacity) and $P_{max} = 0.2$. It is important to note that this embedding method always reduces the robustness of the code, however, this reduction can be precisely controlled by changing the value of the P_{max} parameter and the center size d_a. Smaller center sizes imply a unifrom tone change similar to the results in [14] while larger center regions yield similar results to the method in [5]. The main advantage here is that the user can

select which is best in each case given more versatility for the integration in different designs.

B. Visual Quality of the Embeddings

Fig. 7 depicts the results of embedding a QR code into different images and logos. Each embedding is compared with their corresponding counterparts using the methods in [5] or [14]. Different center sizes and halftone masks are used for each embedding. For an objective evaluation of the quality different metrics are compared in Table I, including the metric J (18) used in the optimization. An interesting feature of the proposed embedding method is the possibility to generate embeddings with different textures by changing the halftone mask as in Fig. 7(b) where a clustered dot mask aligned with the center of the QR modules was used generating pseudo random cluster of dots. This feature also allows to solve the problem of dot gain generated when the embeddings are printed simply by using appropriate halftoning masks.

IX. Discussion and Conclusion

A method to embedded QR codes into color images was presented, allowing to automatically generate embeddings with limited probability of detection error for a given noise power. This method shares different elements with previous methods such as [5], [14], and [15] since it allows to optimize the luminance values of individual pixels in the image at locations selected via halftoning techniques and by setting the centers of the QR modules to specific values. The method presented in [14] change the luminance of modified pixels into two luminance levels analogous to the α and β used here. However these levels are global and the selection of its optimal values is based on the assumption that the image is uniformly distributed and that the binarization threshold is fixed. The method developed here is much more general and includes these as particular cases. We make an accurate approximation of the image statistics using its local mean and variance and develop a model for the probability of error using a common binarization algorithm widely used in real applications. The proposed method also introduces a novel technique to distribute modified pixels based on halftoning methods which help to reduce the visual impact of the modification. The embedding of halftones into QR codes was proposed in [15] where the location of binary pixels in the QR modules is optimized to maximize visual quality and decoding robustness. This method however is only suitable to generate binary halftones and no modification for the embedding of color or multilevel halftones was proposed.

In addition to the use of halftones to diminish the visual distortion of the embedding, the method presented here defines a quality metric which considers color, tone and structural similarity used to select the optimal luminance of modified pixels. To fully leverage the characteristics of QR decoders, central pixels in the QR modules play an important role in our algorithms similarly to [5] and [15]. However in contrast to these methods which fix the ratio between central and non central pixels, the method proposed here allows to choose the number of central pixels and then optimize the location and luminance of modified pixels to achieve particular error limits. In a similar way as in [15] or [12] different regions of the image are prioritized by using local probability limits $P_{max,i}$. These bounds can be used to select the codewords in error allowing to distribute the errors spatially to regions where the image presents non relevant features.

The main limitation of this method is the assumption that single pixels in the QR code can be resolved in the acquired image and that they are only corrupted by Gaussian noise. This is a simple yet powerful model but it adds a constraint on the minimum resolution of the image. Different trials using mobile devices and printed codes however have shown that for common scanning distances the models are indeed applicable and the decoding rates are reasonable in comparison with similar methods while having a noticeable improvement in visual quality with respect to the metric defined. An important factor for the success of the decoding is the correct estimation of the variance of the noise to accommodate different illuminations or camera blur. The values of σ_η used in this paper are between 0.1 and 0.2 and were selected by experimentation with different images and standard fluorescent illumination in an office environment. It is important to note that the embedding method might fail to decode if the reading application uses a different binarization method than the one assumed here, however since the Zxing library is widely adopted this is not a severe limitation. Finally as a consequence of the elaborate models for the visual quality and the probability of error the implementation of the method is significantly more computationally intensive than other methods. The present implementation in Matlab takes 30 minutes to optimize a color image of size 350×350 pixels, but more efficient implementa-tions that take advantage of parallelism and especial processors like GPUs might reduce the optimization time to practical levels.

Appendix A
Probability of Binarization Error

If we assume that all pixels in the window are indistinguishable then we can treat them as non-central pixels and assume that they can only take two independent values α or β. The binarization threshold at location (i, j) is in general given by (2) and it can be divided in three components $t_{i,j} = a_{i,j} + b_{i,j} + \eta_{i,j}$ with

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

$$b_{i,j} = \frac{1}{25 \times 64} \sum\nolimits_{p,q} \sum_{\substack{(k,l)\in B_{p,q} \\ (k,l)\notin \mathcal{C}}} Y[k,l]. \tag{22}$$

Given the set of modified pixels these components are independent and is possible to evaluate the distribution of the threshold as the convolution of the distributions of its components. The component $a_{i,j}$ can be modeled as a binomial distribution assuming independence of the code pixels given by

$$P(a_{i,j} = x) = \sum_{k=0}^{n} \delta\left(x - \left[\frac{k\beta}{25 \times 64} + \frac{(n-k)\alpha}{25 \times 64}\right]\right) \binom{n}{k} p_1^k p_0^{n-k} \tag{23}$$

where k is the number of pixels with modified luminance β 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

$$\mu_Y = \frac{E[Y_{k,l}](25 \times 64 - n)}{25 \times 64} = E[Y_{k,l}](1 - p_c)$$

and variance $\sigma_Y^2 = Var[Y_{k,l}]\frac{1-p_c}{25\times 64}$. Combining these expressions and using the fact that $a_{i,j}$, $b_{i,j}$ and $\eta_{i,j}$ are independent give $\mathcal{C}$, the probability distribution of the threshold $t_{i,j}$ is given by

$$\begin{aligned} P(t_{i,j} = t) &= \sum_{k=0}^{n} (f_Y * f_\eta)(t - t_k) \binom{n}{k} p_1^k p_0^{n-k} \\ &= \sum_{k=0}^{n} f(t - t_k) \binom{n}{k} p_1^k p_0^{n-k} \end{aligned} \quad (24)$$

where

$$t_k = \left\lceil \frac{k\beta}{25 \times 64} + \frac{(n-k)\alpha}{25 \times 64} \right\rceil$$

are the possible outcomes of the sum of n modified pixels. The distribution $f_\eta(n)$ corresponds to the Gaussian noise and $f = f_Y * f_\eta$ is the convolution of the average image values and noise distributions. Since Y and n are independent and Gaussian, f is also Gaussian with mean $\mu = \mu_Y$ and variance

$$\sigma = \sqrt{\sigma_Y^2 + \sigma_\eta^2}.$$

Finally the conditional probabilities on (7) are obtained by conditioning on the set of modified pixels I_{p_c} as follows

$$\begin{aligned} P(Y_{i,j}^{out} > t_{i.j} | q_{i,j} = 0) = {} & P(\alpha > t_{i.j}) p_c \\ & + P(Y_{i,j} > t_{i.j})(1 - p_c) \end{aligned} \quad (25)$$

and

$$\begin{aligned} P(Y_{i,j}^{out} < t_{i.j} | q_{i,j} = 1) = {} & P(\beta < t_{i.j}) p_c \\ & + P(Y_{i,j} < t_{i.j})(1 - p_c). \end{aligned} \quad (26)$$

Using (25), (26) and substituting in (6) we obtain the probability of binarization error

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

where F is the cumulative distribution function of f and

$$w_k = \binom{n}{k} p_1^k p_0^{n-k}.$$

The probability of error depends on the distribution of the image, the QR code, the transformation parameters α, β, p_c and the noise power level σ_η.

REFERENCES

[1] *Information Technology—Automatic Identification and Data Capture Techniques—Bar Code Symbology QR Code*, Int. Org. Standardization, Geneva, Switzerland, ISO/IEC 18004, 2006.

[2] M. Hara, M. Watabe, T. Nojiri, T. Nagaya, and Y. Uchiyama, "Optically readable two-dimensional code and method and apparatus using the same," U.S. Patent 5 726 435, Mar. 10, 1998.

[3] T. J. Soon, and Q. R. Code. Synthesis Journal, 2008, section 3 [Online]. Available: http://www.itsc.org.sg/pdf/synthesis08/Three_QR_Code.pdf

[4] R. Cox. (2012, Apr.). *Qart Codes* [Online]. Available: http://research.swtch.com/qart

[5] Visualead company, Herzelia, Israel. (2013, Jan.). *Free Visual QR Code Generator* [Online]. Available: http://www.visualead.com/

[6] S. Ono, K. Morinaga, and S. Nakayama, "Two-dimensional barcode decoration based on real-coded genetic algorithm," in *Proc. IEEE CEC*, Jun. 2008, pp. 1068–1073.

[7] D. Samretwit and T. Wakahara, "Measurement of reading characteristics of multiplexed image in QR code," in *Proc. 3rd Int. Conf. Intell. Netw. and Collaborative Syst.*, 2011, pp. 552–557.

[8] T. Wakahara and N. Yamamoto, "Image processing of 2-dimensional barcode," in *Proc. 14th Int. Conf. Network-Based Inform. Syst.*, 2011, pp. 484–490.

[9] Program, information storage medium, two-dimensional code generation system, image generation system and printed material, by S. Oouchi. (2007, Mar. 1). *Patent 7950589* [Online]. Available: http://www.google.com/patents/US7950589

[10] M. Hagiwara. (2013). *Qrjam* [Online]. Available: http://qrjam.jp/

[11] K. Fujita, M. Kuribayashi, and M. Morii, "Expansion of image displayable area in design QR code and its applications," in *Proc. Forum Inform. Technol. Papers*, 2011, vol. 10, no. 4, pp. 517–520.

[12] Y. Lin, Y. Chang, and J. Wu, "Appearance-based QR code beautifier," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2198–2207, Dec. 2013.

[13] D. W. Inc., Elkhart, IN, USA. (2013). *Logoq* [Online]. Available: http://www.qrcode.com/en/codes/logoq.html

[14] Z. Baharav and R. Kakarala, "Visually significant QR codes: Image blending and statistical analysis," in *Proc. IEEE ICME*, Jul. 2013, pp. 1–6.

[15] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra, "Halftone QR codes," *ACM Trans. Graph.*, vol. 32, no. 6, p. 217, 2013.

[16] O. Bulan, H. Blasinski, G. Sharma, "Color QR codes: Increased capacity via per-channel data encoding and interference cancellation," in *Proc. 19th CIC*, San Jose, CA, USA, 2011.

[17] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, nos. 285–296, pp. 23–27, 1975.

[18] A. Brink and N. Pendock, "Minimum cross-entropy threshold selection," *Pattern Recognit.*, vol. 29, no. 1, pp. 179–188, 1996.

[19] J. Sauvola and M. Pietikäinen, "Adaptive document image binarization," *Pattern Recognit.*, vol. 33, no. 2, pp. 225–236, 2000.

[20] J. Zhou, Y. Liu, and P. Li, "Research on binarization of QR code image," in *Proc. ICMT*, 2010, pp. 1–4.

[21] H. Yang, A. Kot, and X. Jiang, "Binarization of low-quality barcode images captured by mobile phones using local window of adaptive location and size," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 418–425, Jan. 2012.

[22] S. Owen. (2012). *Zxing, Multi-Format 1d/2d Barcode Image Processing Library With Clients for Android, Java and C++* [Online]. Available: https://code.google.com/p/zxing/

[23] D. L. Lau and G. R. Arce, *Modern Digital Halftoning*. New York, NY, USA: Marcel Dekker, 2001.

[24] R. Floyd and L. Steinberg, "An adaptive technique for spatial grayscale," in *Proc. Soc. Inform. Display*, vol. 17. 1976, pp. 78–84.

[25] R. L. Stevenson and G. R. Arce, "Binary display of hexagonally sampled continuous-tone images," *J. Opt. Soc. Amer. A*, vol. 2, no. 7, pp. 1009–1013, 1985.

[26] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital halftoning by means of green-noise masks," *J. Opt. Soc. Amer. A*, vol. 16, no. 7, pp. 1575–1586, 1999.

[27] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Green-noise digital halftoning," *Proc. IEEE*, vol. 86, no. 12, pp. 2424–2444, Dec. 1998.

[28] G. J. Garateguy, G. R. Arce, and D. L. Lau, "Voronoi tessellated halftone masks," in *Proc. 17th IEEE ICIP*, Sep. 2010, pp. 529–532.

[29] M. Analoui and J. P. Allebach, "Model-based halftoning using direct binary search," in *Proc. SPIE/IS&T Symp. Electron. Imag., Sci. Technol., Int. Soc. Opt. Photon.*, 1992.

[30] R. Ulichney, "The void-and-cluster method for generating dither arrays," *Proc. SPIE, Human Vis., Vis. Process., Digit. Display IV*, vol. 1913, pp. 332–343, 1993.

[31] D. L. Lau, R. Ulichney, and G. R. Arce, "Blue and green noise halftoning models," *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 28–38, Jul. 2003.

[32] J. B. Rodriguez, G. R. Arce, and D. L. Lau, "Blue-noise multitone dithering," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1368–1382, Aug. 2008.

[33] H. J. Trussell, E. Saber, and M. Vrhel, "Color image processing [basics and special issue overview]," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 14–22, Jan. 2005.

[34] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009.

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[36] R. Nasanen, "Visibility of halftone dot textures," *IEEE Trans. Syst., Man, Cybern.,Syst.*, vol. SMC-14, no. 6, pp. 920–924, Nov./Dec. 1984.

[37] W.-M. Pang, Y. Qu, T.-T. Wong, D. Cohen-Or, and P.-A. Heng, "Structure-aware halftoning," *ACM Trans. Graph.*, vol. 27, no. 3, p. 89, 2008.

[38] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

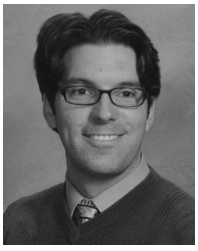
[39] R. H. Byrd, J. C. Gilbert, and J. Nocedal, "A trust region method based on interior point techniques for nonlinear programming," *Math. Program.*, vol. 89, no. 1, pp. 149–185, 2000.

[40] Mathworks, Natick, MA, USA. *Fmincon Reference Page Documentation* [Online]. Available: http://www.mathworks.com/help/optim/ug/fmincon.html

[41] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital color halftoning with generalized error diffusion and multichannel green-noise masks," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 923

Gonzalo J. Garateguy (S'06) received the B.E.E. degree from the Catholic University of Uruguay, Montevideo, Uruguay, in 2007, and the Ph.D. degree from the University of Delaware, Newark, DE, USA, in 2014. He is currently a Signal Processing Quality Engineer with The MathWorks, Inc., Natick, MA, USA. His research interests include signal and image processing, halftoning, and convex optimization.

Gonzalo R. Arce (F'00) received the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA. He is the Charles Black Evans Professor of Electrical and Computer Engineering with the University of Delaware, Newark, DE, USA. He was the Fulbright-Nokia Distinguished Chair of Information and Communications Technologies at Aalto University, Helsinki, Finland, in 2010. His research interests include statistical signal processing and computational imaging. He has co-authored the texts *Nonlinear Signal Processing* (Wiley, 2004), *Modern Digital Halftoning* (CRC Press, 2008), and *Computational Lithography* (Wiley, 2010). He has served as an Associate Editor of several journals of the IEEE, OSA, and SPIE.



Daniel L. Lau (SM'09) received the B.Sc. (Hons.) degree from Purdue University, West Lafayette, IN, USA, in 1995, and the Ph.D. degree in electrical engineering from the University of Delaware, Newark, DE, USA, in 1999. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY, USA, where his early research interest focused on digital printing and halftoning. He has authored *Modern Digital Halftoning*. His current research interests include 3-D measurements by means of structured light, with his contributions having been featured in numerous machine vision trade magazines, and the creation of a startup corporation, i.e., Seikowave Inc., Lexington. Apart from the research, he is also an Avid Digital Photographer primarily interested in lenticular photography. He has authored papers in the PROCEEDINGS OF THE IEEE.

Ofelia P. Villarreal received the B.E.E. degree at the University of Valle in Cali, Colombia, in 2006. She is currently a Master student at the Department of Electrical and Computer Engineering at the University of Delaware, Newark, DE, USA. Her current research interests include signal and image processing.