VISUALLY SIGNIFICANT QR CODES: IMAGE BLENDING AND STATISTICAL ANALYSIS

Zachi Baharav

Corning West Technology Center
Palo Alto, California
baharavi@corning.com

Ramakrishna Kakarala

School of Computer Engineering
Nanyang Technological University
ramakrishna@ntu.edu.sg

ABSTRACT

QR codes are widely used as a means of conveying textual information, such as emails, hyperlinks, or phone numbers, through images that are interpreted using a smartphone camera. The codes take up valuable space in print media. The random appearance of QR codes not only detracts from the production values of the advertisement in which they appear, but the codes are also visually insignificant in the sense that a human cannot discern the vendor, brand, or purpose of the code just by looking at it, without the aid of scanning software. Though neither the aesthetics nor the visual significance of the code matter for scanning purposes, they do matter for advertising layout and, more importantly, can provide valuable brand distinction. In this paper, we show how the visually significant QR codes may be obtained by image blending. Unlike various ad-hoc methods that have been proposed by others, our method leaves completely intact the error correction budget of the code. Our method allows images as diverse as corporate logos and family photographs to be embedded in the code in full color. We provide a detailed statistical analysis of the method to show the effect of blending on error rates in noisy environments.

Index Terms— QR codes, Image Fusion, Statistical Analysis

1. INTRODUCTION

The purpose of this paper is to describe a statistically-analyzed method for blending color images into a QR code to improve its visual significance. The QR (for "quick response") code is a type of two-dimensional (2-D) barcode that was originally developed by the Denso Wave corporation, and subsequently became an international standard [1]. The code is easily decoded using image analysis techniques, and decoders are now present in many camera-equipped phones. Consequently, the QR code is widely used, appearing in newspapers, magazines, and billboards. QR codes may be considered a major success story of modern image processing, and provide an excellent illustration of the power of image analysis techniques.

A QR code is a high-contrast black and white code, as illustrated in Figure 1(a). The code appears random aside from the corner markers, which provide landmarks for finding the code. The random appearance of QR codes is not visually appealing, and significantly distracts from the overall production quality of the advertisements in which they appear. More importantly, QR codes are not *visually significant*, by which we mean that a person cannot tell the purpose of the code just by looking at it, without resorting to decoding with scanning software. An effort at making the code visually significant is shown in Figure 1(b), in which text has been inserted by simply replacing the corresponding portion of the code. Though the image in Fig. 1(b) scans correctly, it loses error resilience because the word's insertion replaces roughly 20% of the code. Those changes introduce bit errors that are overcome by the redundancy and error correction built into the code. The drawback of this replacement method is that the error budget of the original code in Fig. 1(a), which is rated at 30%[1], has been severely depleted. That reduces the code's effectiveness in print media which are subject to smudges, wrinkles, and other sources of error. Figure 1(c) shows the result using the method proposed in this paper, which relies on image blending to alter only the appearance of the code without altering the decoding. Most importantly, our method provides the same level of visually significance while leaving the error budget completely intact. As we show in this paper, the proposed method is sufficiently general to allow full color blending of a variety of photographs, graphics, and text to improve the appearance and significance of the code without sacrificing error correction.

1.1. Literature review

QR codes are among the most popular of several 2-D code formats in use today, a list of which includes the HCCB (also known as Microsoft Tag©), the data matrix, and the Aztec code. The motivations behind the design of QR codes, as described in patents [2] and standards[1], are as follows: (a) to provide a 2-D code combining both high data-bearing area and error correction; (b) to allow easy detection of rotation

[1]Source: `www.denso-wave.com/qrcode/qrfeature-e.html`

(a) (b) (c)

Fig. 1: (In color) Examples of 3 QR codes, each of which decodes to the URL of a journal. Image (a) is the original code; (b) shows the code with text inserted to provide visual significance, at the expense of reducing error budget; the rightmost image (c) shows the result using the methods of this paper, which provides the same visual significance without losing any of the error correction capability.

angle and processing of codes at arbitrary orientation to the camera; (c) to allow scalability to versions, each containing progressively higher rates of data and error correction, as well as auxiliary information such as timing patterns to facilitate decoding. The basic format of a QR code is described in several sources [3]. There are 40 different versions of the QR code, each higher version containing 4 modules (a module is the smallest black or white square) more on a side than the next lower one and hence being capable of carrying more information; with version 1 having 21 modules on a side, version 40, for example, has 177 modules. The codes allow a choice of 4 levels of error correction, with the highest level being capable of 30% error restoration.

Image processing researchers have explored various aspects of QR codes. We present a review here to show the context of QR codes in vision, though the papers themselves are not concerned with visual significance or aesthetics. Belussi & Hiram [4] develop a boosted classifier, similar to the Viola-Jones face detector, to improve detection of the position markers in cluttered background. Colored QR codes have been proposed for the purpose of increasing data rate; although color has the potential to improve appearance and significance, those aspects have not been explored. The previous work on color codes has explored robust techniques for locating HCCB codes[5], as well as choice of color mapping to minimize cross-channel interference [6]. There are, in addition, numerous works that are not relevant to this paper on using QR codes to solve vision problems of locating objects, for example by using them as markers.

Previous work in enhancing the appearance of QR codes takes advantage of the built-in error correction. As exemplified in Fig. 1(b), a number of modules as determined by the error correction level are deliberately altered at will, to create a pattern that looks more attractive and visually significant, yet decodes correctly. There are two disadvantages to that approach. First, beyond simple replacement, there is as yet no automated method that is available for choosing which modules to modify, and second, the error correction budget being drawn down makes the code vulnerable to decoding errors in printed media such as stains or wrinkles. The method that we propose in this paper is both automated and also leaves the error budget completely intact.

2. METHOD FOR IMAGE BLENDING

The decoding of a QR code is relatively simple. An image of the code is obtained from a (color) smartphone camera. In a typical decoder, such as for example, the popular Google ZXing[2], the image from the camera is converted from RGB color space to luminance. The chrominance components of the image are ignored. In the luminance channel, the pixel values are compared to a threshold λ. Pixels whose luminance exceeds λ are deemed to belong to a white module, and those below λ are considered black.

This simple methods of decoding allows for some latitude in QR code design. If we treat luminance values as normalized to the interval $[0, 1]$, then sensed values in the range $(\lambda, 1]$ are considered white by the decoder, and those in the interval $[0, \lambda]$ are considered black. Therefore, we may in theory modify the QR code source pixels so that pixels in a white module are transformed from white to any RGB coordinate whose luminance value exceeds λ, without creating a decoding error; similarly, we can modify black modules of the QR source so that their luminance falls below λ. In practice, the luminance sensed by the camera fluctuates due to lighting conditions and noise. Therefore, it is prudent to use upper and lower modification thresholds denoted τ_u, τ_l, respectively, so that white pixels are modified to have luminance τ_u where $\tau_u > \lambda$, and black pixels to have luminance $\tau_l < \lambda$. The differences $\tau_u - \lambda$ and $\lambda - \tau_l$ provide a margin to reduce decoding error. Below, we analyze statistically the effect of threshold choices on error.

[2] http://code.google.com/p/zxing/

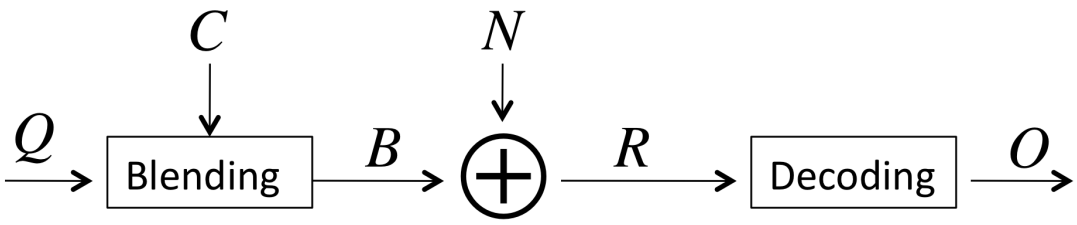


Fig. 2: The three stages involved in QR code blending and sensing are shown. The blending stage combines the color image C and the QR code image Q based on the luminance of C and the binary value of Q. Transmission and scanning introduce noise, which is modeled as N added to B. Decoding compares the received code pixel R to a threshold, and outputs a binary image O.

The latitude in decoding allows blending of a color image C into a QR code, denoted Q, without incurring an decoding error. Figure 1(c) shows an example of a blended image. For every pixel in the color image with color coordinates RGB, let Y denote its luminance, obtained through the transformation

$$Y = 0.29 * \text{Red} + 0.59 * \text{Green} + 0.12 * \text{Blue}. \quad (1)$$

The blending of C and Q to produce an output B is accomplished by replacing pixels of Q with those of C. We assume that pixels of Q are normalized so that white pixels have a luminance of 1, and black pixels have a luminance of 0. Let C_Y denote the luminance of image C, and let B_Y denote the luminance of the blended image B. Initially, set $B_Y = Q$, so that the output image matches the QR code unless modified. At each pixel location (i, j), the luminance $B_Y(i, j)$ is assigned as follows.

if $Q(i, j) = 1$ **then**
  $B_Y(i, j) \leftarrow \max\{C_Y(i, j), \tau_u\}$
else
  $B_Y(i, j) \leftarrow \min\{C_Y(i, j), \tau_l\}$
end if

Intuitively, when $Q = 1$, the algorithm ensures that bright pixels (above τ_u) are used directly in the blended output, while dimmer pixels below τ_u are clipped at τ_u, and similarly for $Q = 0$. Since only luminance of B matters for decoding, the chrominance of B is assigned that of C without modification. The algorithm ensures that the blended output image B preserves the bright part of the input image C when $Q = 1$, and the dark part of output image C when $Q = 0$, since those will be above and below τ_u, and τ_l, respectively. The middle luminance regions, where $\tau_l \leq C_Y \leq \tau_u$, are essentially boosted to appear bright when $Q = 1$ and dark when $Q = 0$, respectively. Therefore the middle regions essentially match the appearance of the code. Note that in the limiting case of $\tau_u = 1$, $\tau_l = 0$, luminance blending does not occur, i.e., $B_Y = Q$. We refer to this below as the "without blending" case.

The pixel-based algorithm may be improved to gain more latitude in blending by noting that QR codes are decoded a module at a time. Each module contains $N \times N$ pixels. The decoder averages the luminance of the QR image over a module; therefore, if the average luminance over $N \times N$ pixels exceeds λ, it is decoded as 1, otherwise, it is decoded as 0. We modify the algorithm described above to use this additional freedom in blending as follows. Let $\overline{C_Y}(i, j)$ denote the average luminance of the color image in pixels that correspond to the (i, j)-th module, and let $Q(i, j)$ denote the luminance of module at (i, j) in the code image. Let $B_Y(i, j) \leftarrow C_Y(i, j)$ denote the replacement of all the $N \times N$ pixels in the module for B_Y by their corresponding values in C_Y. The blending of C and Q is described in Algorithm 1.

Algorithm 1 Rules for blending a QR code with an image on a block-by-block basis

if $Q(i, j) = 1$ **then**
  if $\overline{C_Y}(i, j) > \tau_u$ **then**
    $B_Y(i, j) \leftarrow C_Y(i, j)$
  else
    $B_Y(i, j) \leftarrow \min\{C_Y(i, j) + \left(\tau_u - \overline{C_Y}\right), 1\}$
  end if
else
  if $\overline{C_Y}(i, j) < \tau_l$ **then**
    $B_Y(i, j) \leftarrow \min\{C_Y(i, j), \tau_l\}$
  else
    $B_Y(i, j) \leftarrow \max\{C_Y(i, j) + \left(\overline{C_Y} - \tau_l\right), 0\}$
  end if
end if

An intuitive description of the block-based blending is that, when the mean luminance of the color image falls below the threshold (for $Q = 1$ case), we add enough to each pixel to raise the mean above threshold, making sure that luminances are clipped to 1 if they exceed full scale; similarly for the $Q = 0$ case.

Note that, due to the clipping necessary to limit luminance to $[0, 1]$, there is a slight chance that we may obtain a decoding

error due to the mean of B_Y not being above τ_u when $Q = 1$, and similarly for $Q = 0$. Therefore, it is prudent to set the blending thresholds τ_u, τ_l to be significantly above, and below, respectively the decoding threshold λ. We completely avoid this risk in practice by checking the mean of B_Y after clipping, and adjusting it by a small amount ϵ until the mean exceeds τ_u for $Q = 1$ or τ_l for $Q = 0$. In the next section, we provide a detailed statistical analysis showing the error probability in noisy transmission resulting from a choice of the blending thresholds.

3. STATISTICAL ANALYSIS OF ALGORITHM

In order to understand the effect of choosing blending thresholds, we analyze the transmission and reception model using the diagram shown in Figure 2. We assume that the camera's automatic white balance is operating correctly, so that we may neglect the illuminant color temperature and assume that white appears (roughly) as white. The decoding stage shown in that figure must output either a 1 or 0. We model its operation with a comparison of R to a fixed threshold:

$$O = \begin{cases} 1 & \text{if } R \geq \lambda, \\ 0 & \text{else} \end{cases} \tag{2}$$

The probability of error is

$$\begin{aligned} P(\text{Err}) \quad = \quad & P\{O = 1|Q = 0\}\, P\{Q = 0\} + \\ & P\{O = 0|Q = 1\}\, P\{Q = 1\} \end{aligned} \tag{3}$$

Assume that $P\{Q = 0\} = P\{Q = 1\} = 0.5$ for the QR code. Assuming that transmission, scanning, and lighting variations result in an additive noise N as in Figure 2, the key element in the first term may be expanded as

$$P\{O = 1|Q = 0\} = P\{B_Y + N \geq \lambda|Q = 0\} \tag{4}$$

Using the first method of image blending described in the previous section, the right hand side becomes

$$\begin{aligned} & P\{C_Y + N \geq \lambda|Q = 0 \,\&\, C_Y < \tau_l\}\, \alpha + \\ & P\{\tau_l + N \geq \lambda|Q = 0 \,\&\, C_Y \geq \tau_l\}\, (1 - \alpha), \end{aligned} \tag{5}$$

where

$$\alpha = P\{C_Y < \tau_l\}. \tag{6}$$

A similar expansion can be carried out for $P\{O = 0|Q = 1\}$. We obtain insight into (5) by assuming plausible models for C_Y and N. Assume that C_Y is normalized luminance that is uniformly distributed in $[0, 1]$, and assume that N is independently distributed as $\mathcal{N}(0, \sigma)$. Using the fact that the standard deviation of a uniform $[0, 1]$ variable is $1/\sqrt{12}$, the signal-to-noise ratio (SNR) is then

$$\text{SNR} = 20 \log_{10} \frac{1}{\sigma\sqrt{12}} \tag{7}$$

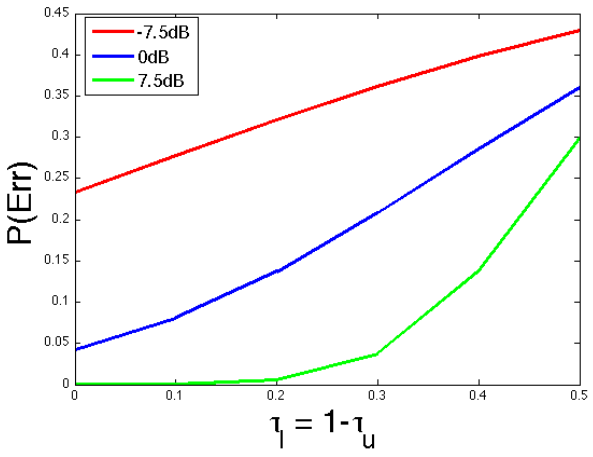


Fig. 3: (In color) Plot of the error probability as a function of symmetric thresholds $\tau_l = 1 - \tau_u$, for $\lambda = 0.5$. The three different curves represent three different SNRs as shown in the legend; the red curve, for SNR $= -7.5$ dB, is essentially a straight line.

Using those models, we have from (6) that $\alpha = \tau_l$. Furthermore, note that the pdf of $z = u + n$, where u is distributed uniformly in $[a, b]$, and n is an independent random variable that is distributed as a Gaussian $\mathcal{N}(0, \sigma)$, is easily derived to be the function

$$f_z(x; a, b) = \frac{\Phi(\frac{b-x}{\sigma}) - \Phi(\frac{a-x}{\sigma})}{b - a}, \tag{8}$$

where Φ is the cumulative distribution of $\mathcal{N}(0, 1)$. The limiting case $a = b$, which entails that u is a constant, is that z is a Gaussian with mean a, i.e., $z \sim \mathcal{N}(a, \sigma)$. Let $F_z(x; a, b, \sigma)$ denote the cumulative distribution of (8). We find that

$$\begin{aligned} P\{O = 1|Q = 0\} \quad = \quad & [1 - F_z(\lambda; 0, \tau_l, \sigma)]\, \tau_l \\ & + \left[1 - \Phi\left(\frac{\lambda - \tau_l}{\sigma}\right)\right](1 - \tau_l). \end{aligned} \tag{9}$$

Using a symmetric approach, we find that

$$\begin{aligned} P\{O = 0|Q = 1\} \quad = \quad & F_z(\lambda; \tau_u, 1, \sigma)(1 - \tau_u) \\ + \quad & \Phi(\frac{\lambda - \tau_u}{\sigma})\tau_u. \end{aligned} \tag{10}$$

The probability of error is obtained from (3) by summing equations (9) and (10), each multiplied by 0.5. For the special case of $\tau_u = \tau_l = \lambda = 0.5$, and standard deviation $\sigma = 1/\sqrt{12}$, we obtain after inserting those numbers into the respective equations that, to two decimal places, $P(\text{Err}) = 0.36$. This high probability of error reflects both the thresholds used, which result in more of the image being blended, and the low SNR of 0dB. Note that a QR code contains considerable error correction through the use of a Reed-Solomon code, which ameliorates the effect of bit errors. We

Table 1: Comparision of $P(\text{Err})$ vs SNR with blending, using $\tau_u = \tau_l = 0.5$, and without blending (no image used, so that $B = Q$).

SNR (dB)	$P(\text{Err})$ with blending	$P(\text{Err})$ without blending
-15	0.47	0.38
-7.5	0.43	0.23
0	0.36	0.04
7.5	0.29	0.00
15	0.27	0.00

compare the value $P(\text{Err}) = 0.36$ to the error probability when no blending is employed, which is obtained by setting $\tau_l = 0$, and $\tau_u = 1$. For the same SNR of 0dB, we obtain $P(\text{Err}) = 0.04$. Therefore, the cost of blending with thresholds of $\tau_l = \tau_u = 0.5$ is an increase in $P(\text{Err})$ of 0.32. Note that the error is due only to the SNR, and not to to the image blending since that does not create additional decoding mistakes in the noise-free case. Table 1 shows the results for different values of SNR.

Table 2: Experimental conditions under which tests were conducted.

SNR (dB)	-7.5, -10.0, -12.5
Ilumination (lux)	10, 200, 400
Surface	Wall (projected), LCD Display, Laptop display, Printed paper
Camera	Android© phone iPhone©, iPad©

We may reduce the error by setting τ_u, τ_l to be above and below λ, respectively. To gain more insight into the contributing factors, we computed the error probability using (9), (10) and (3), for symmetric thresholds $\tau_u = 1 - \tau_l$. Figure 3 plots the results for three different SNR levels. In particular, we see that for SNR of -7.5dB (red curve), the result is well approximated by a linear fit:

$$P(\text{Err}) \approx 0.4\tau_l + 0.24. \tag{11}$$

This simple expression provides a convenient way to estimate the resilience of embedding in a noisy environment. For example, with this very low SNR, we can keep the error probability below 0.3, which is the maximum that can be fixed by the QR code's error correction, by setting $\tau_l \leq 0.15$. Note that if we set $\tau_l = 0$, which is the case of no blending, then equation (11) estimates the error probability at 0.24. The error level may be understood by noting that SNR is assumed to be low, at -7.5dB, in order to derive (11), and moreover, the decoding threshold is $\lambda = 0.5$.

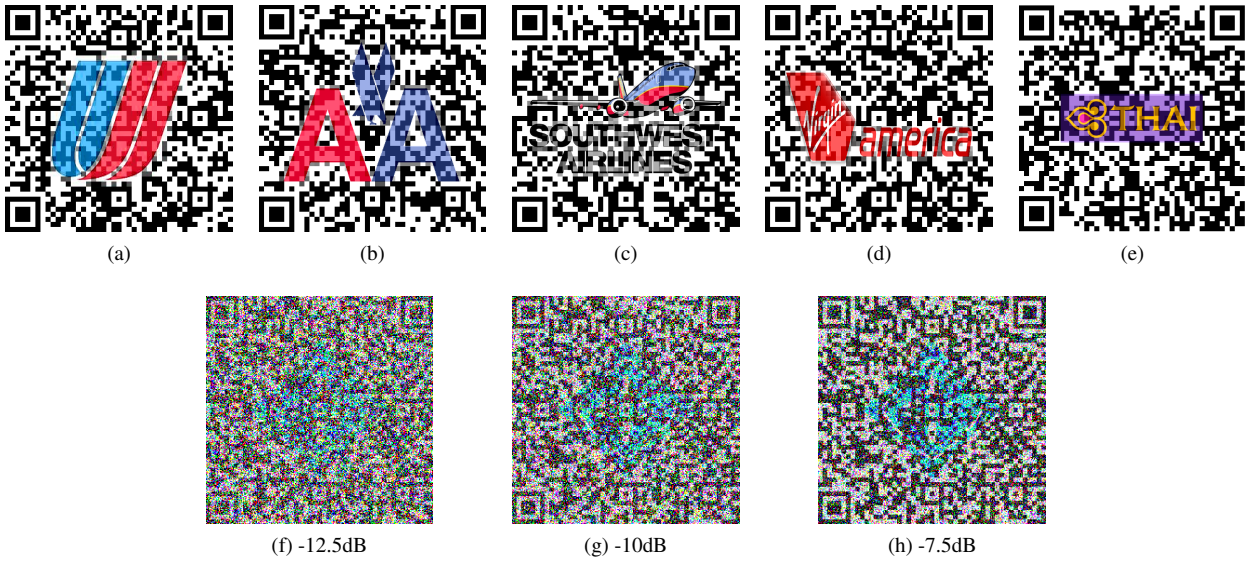
4. RESULTS

QR codes are frequently placed where they indicate the vendor as well the object. For example, they appear in electronic boarding passes, product warranties, and garment use and care labels. However, the code does not visually identify the vendor, and therefore reduces distinctiveness of the product. Figure 4 shows the result of blending a color image with a QR code with the block-based algorithm described in Section 3. We use a threshold of $\tau_l = 0.32$ and $\tau_u = 1 - 0.32 = 0.68$ in producing the images for this figure. Furthermore, the base codes used the following QR settings: version $= 7$, an error correction level of high (H), which as previously noted can restore up to 30% of the codewords. The QR codes were scaled up from 45×45 to 1450×1450 pixels. The visual significance of the blended codes shown in Fig. 4 is obvious from the immediate identification of the airline. Note that in creating codes through blending, we have not given up any of the 30% error correction budget of the original codes.

It is important to note that our blending method is fully compatible with existing decoders. We tested decoders on four different cameras, including two smartphones, and two models of iPads©, and found that they were all able to decode the images in Figure 4 correctly. To test noise resilience, we added Gaussian noise to the code of Fig. 1(c) at three different SNR levels as defined by (7). Figure 4 shows examples of the three SNRs tested. We tested the decodeability with four different cameras, and in two low light environments: a darkened conference room without windows, and an office interior using only dim exterior light. The illumination levels ranged from 10 lux to 400 lux. In each trial, we took the following steps to ensure a fair test. First, we positioned the camera near enough the surface to ensure that the code, surrounded by a white border, completely filled the live preview window of the smartphone app, so that only the code is being interpreted. Second, we made sure that the code was in focus by letting the autofocus stabilize. Third, we made sure that the corner finder patterns [3] were located by the app; those patterns are indicated by visible markers on the preview screen. The experimental conditions are tabulated in Table 2.

In each case, we were able to decode the image at SNR $= -7.5$dB reliably, but not images at lower SNRs. We repeated the tests with two of the images (Thai Airways and EVA Airways) from Fig. 4, with similar results. The blended codes in this experiment used $\tau_l = 0.15$. From (11) we see, for this τ_l and -7.5 dB SNR, that $P(\text{Err}) \approx 0.3$, a number that matches the 30% error correction capacity of the codes used. These tests supports the use of the analysis in Section 3 to predict error resilience of the blending algorithm.

Note that, in the resilience experiment illustrated in Fig. 4, the source of error is the added noise, not the blending. The



(a) (b) (c) (d) (e)

(f) -12.5dB (g) -10dB (h) -7.5dB

Fig. 4: (In color) Logos of airlines are shown embeded into the QR codes for their websites, using the algorithms of this paper. The identity of the airline is visually clear from the code.

Reed-Solomon (RS) code used in the QR pattern is preserved completely intact after blending, and can correct up to 30% decoding errors. An analogy may be useful to understand this effect: writing on a compact disc with green ink leaves intact the disc's RS code, because the red laser used in the player does not see it. However, adding noise on top of the ink leads to more errors than if no ink were used, because the tolerance is now lower. Similarly, it is fair to say that our method of image blending preserves the error correction budget of the code completely intact.

5. CONCLUSIONS AND FUTURE RESEARCH

The modification of QR codes to improve their visual significance is now widespread. Examples of such modifications may be found easily using a web search on terms such as "artistic QR codes", and include rounding corners of the QR modules, embedding images (as in Figure 1(b)), using a mosaic in place of modules, and so on. All of these methods either deliberately use up some of the error budget provided in the code, or are based on heuristics without any analysis of the modification's effect on error. We present in this paper a novel method of blending a color image into the QR code, which in the noise-free case entails no loss of error resilience. Furthermore, we provide a statistical analysis for the noisy case to estimate the increase in error rate due to blending as a function of the thresholds used. In future research, we will examine the construction of blended QR codes with reduced error rate in noisy channels through spatially adaptive blending techniques, including the use of an "alpha" channel to allow selective portions of the code to be blended.

6. REFERENCES

[1] ISO-IEC 18004-2006, "Information technology—automatic identification and data capture techniques — QR code bar code symbology specification," 2006, Available at `www.iso.org`.

[2] M. Hara, M. Watanabe, T. Nojiri, T. Nagaya, and Y. Uchiyama, "Optically readable two-dimensional code and method and apparatus using the same," 1998, US Patent 5726435.

[3] [Online], "QR code," `en.wikipedia.org/wiki/QR_code`, Retrieved 24 October 2011.

[4] L. F. F. Belussi and N. S. T. Hirata, "Fast QR code detection in arbitrarily acquired images," in *Proc. Conference on graphics, patterns and images (SIBGRAPI), 2011*, Maceió, 2011.

[5] Devi Parikh and Gavin Jancke, "Localization and segmentation of a 2d high capacity color barcode," in *Proc. Workshop on applications of computer vision (WACV), 2008*, Copper Mtn, CO, 2008, pp. 1–6.

[6] Orhan Bulan, Henryk Blasinski, and Gaurav Sharma, "Color QR codes: Increased capacity via per-channel data encoding and interference cancellation," in *Proc. Nineteenth Color Imaging Conference (CIC19), 2011*, San Jose, CA, 2011, pp. 156–159.