

Efficient QR Code Beautification With High Quality Visual Content

Shih-Syun Lin, Min-Chun Hu, *Member, IEEE*, Chien-Han Lee, and Tong-Yee Lee, *Senior Member, IEEE*

Abstract—Quick response (QR) code is generally used for embedding messages such that people can conveniently use mobile devices to capture the QR code and acquire information through a QR code reader. In the past, the design of QR code generators only aimed to achieve high decodability and the produced QR codes usually look like random black-and-white patterns without visual semantics. In recent years, researchers have been tried to endow the QR code with aesthetic elements and QR code beautification has been formulated as an optimization problem that minimizes the visual perception distortion subject to acceptable decoding rate. However, the visual quality of the QR code generated by existing methods still leaves much to be desired. In this work, we propose a two-stage approach to generate QR code with high quality visual content. In the first stage, a baseline QR code with reliable decodability but poor visual quality is first synthesized based on the Gauss-Jordan elimination procedure. In the second stage, a rendering mechanism is designed to improve the visual quality while avoiding affecting the decodability of the QR code. The experimental results show that the proposed method substantially enhances the appearance of the QR code and the processing complexity is near real-time.

Index Terms—Gauss-Jordan elimination, quick response (QR) code beautification, Reed-Solomon (RS) code.

I. INTRODUCTION

WITH the increasing popularity of smartphones, QR code (Quick Response Code) has become a popular form for acquiring information of specific object/event in our daily life and numerous applications [1]–[5] are built based on QR codes. The original framework of QR code aims to embed machine-readable messages with error tolerance, and the appearance of the resulting 2D matrix usually looks like a random pattern. Since QR codes are encoded using Reed-Solomon (RS) error-correcting codes, a QR scanner does not have to see every pixel correctly while decoding the content. This error correction

Manuscript received October 13, 2014; revised January 14, 2015 and April 15, 2015; accepted May 19, 2015. Date of publication May 25, 2015; date of current version August 10, 2015. This work was supported in part by the Headquarters of University Advancement, National Cheng Kung University, and in part by the Ministry of Science and Technology of Taiwan under Contract MOST-104-2221-E-006-044-MY3, Contract MOST-103-2221-E-006-106-MY3, and Contract MOST-103-2221-E-006-157-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alessandro Piva.

S.-S. Lin, M.-C. Hu, and T.-Y. Lee are with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan (e-mail: catchylss@gmail.com; anita_hu@mail.ncku.edu.tw; tonylee@mail.ncku.edu.tw).

C.-H. Lee is with MACHVISION Inc., Hsinchu 30075, Taiwan (e-mail: eee1@hotmail.com).

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/TMM.2015.2437711

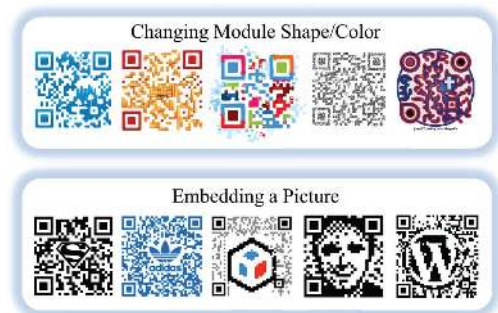


Fig. 1. Two kinds of variations to beautify the QR code.

mechanism makes it possible to introduce a few errors and enables the QR code designer to change the appearance of the QR code to some extent. In the past few years, designers have tried to beautify the QR code by adding aesthetic elements or recognizable visual content [6]. Generally, designers target on two kinds of variants to make the original QR code visually pleasant, i.e. changing the shape/color of modules and embedding a picture into the QR code. Fig. 1 shows some examples of these two kinds of variations. Embedding a picture into the QR code can easily convey the characteristics of the QR code owner and would impress the viewer more. For example, if the QR code is generated for accessing your personal homepage, then embedding your own portrait into it would make the QR code visually unique and representative. Therefore, in this work we aim at the case of embedding a picture to beautify the QR code.

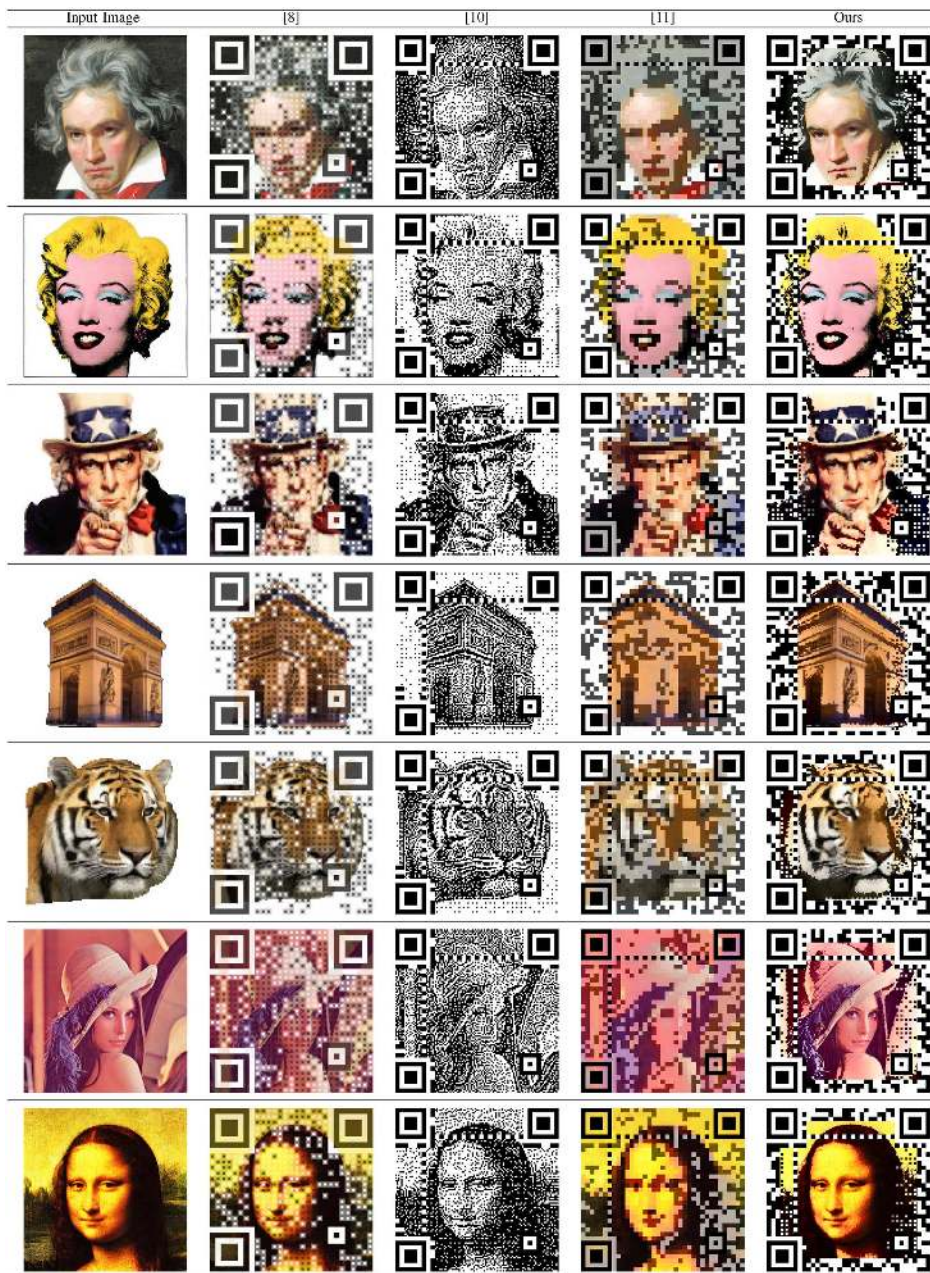
Manually embedding visual content into a QR code without affecting its scanability is not an easy task. Several methods have been proposed to automatically beautify QR code through picture embedding [7]–[9].^{1,2} However, each of them has limitations in terms of the size constraint of the embedded picture, visual quality of the generated QR code, or computational complexity.

Lin *et al.* [7] introduced an error-aware warping technique for deforming the embedded image such that the error in the resulting QR code is minimized and the readability of the code is optimized. Additionally, they reshaped the regular squares of the module by using a binary exemplar to make their local appearances resemble the example shape. Lin's method is applicable when the embedded image is a small logo. Unfortunately, if we hope the embedded visual content occupies a significantly

¹“Unitag QR code generator,” [Online]. Available: <https://www.unitag.io/qr-code>

²“Visualead,” [Online]. Available: <http://www.visualead.com/>

TABLE I
EXAMPLE RESULTS COMPARED WITH DIFFERENT QR CODE GENERATORS



large area in the QR code, the embedded content will be seriously deformed.

Peled *et al.* developed Visualead, a Visual QR Code Generator, which keeps a concentric region of modules unchanged and uniformly blend the neighboring regions with image content. Their key idea is to preserve the original contrast between modules while blending the image content and the QR code. However, their method results in serious artifacts such as corruption at salient regions of the image content (please refer to Table I). Moreover, when the same image content is applied to different QR code, the result quality of the visual QR code varies a lot.

Aiming at retaining local characteristics of the readable QR code while approximating a global shape of the embedded

image content, Chu *et al.* [8] proposed an approach to produce high quality visual QR codes based on the technique of halftone. They built a pattern readability function wherein a probability distribution of what modules can be replaced by which other modules is learnt. Then, the input image is expressed by the learned dictionary to encode the source text. Compared to Visualead, Chu's work produces less corruption since they largely loose the binding between the QR code and the halftone image. But the visual quality of the resulting halftone QR code is still improvable. Lin *et al.* [9] considered both the saliency map and the edge map of the input visual content to optimize the best changeable modules for generating a beautified QR code. The basic unit used for module selection is a "block", which consists of eight modules representing an

8-bits RS codeword. Therefore, the flexibility of modifying module information is constrained by the block size. Moreover, since the arithmetic operations of RS code are defined over a finite field, Lin employed the simulated annealing technique to realize the optimization process rather than directly applying normal convex optimization approaches, and it consequently takes longer time to generate a beautified QR code.

Considering the above-mentioned disadvantages of existing works, we propose a new QR code beautifier and our contributions include the following.

- We propose a two-stage QR code beautifier that not only ensures the decodability but also preserve most visual semantics of the embedded content in near real-time. Compared to existing methods, our work have two merits: 1) the embedded image can occupy large portion of the QR code image. 2) the result QR code has less noise and maintains most visual content of the input image. The proposed QR code beautifier is evaluated on a wide variety of visual content and the results show that our method outperforms the state-of-the-art methods.
- We apply the concept of control bits proposed by Cox [10] to ensure the decodability of the generated QR code and further modify Cox's method by changing the policy of selecting control bits to improve the visual quality.
- We design a rendering method based on an ellipse alpha mask, which makes the QR code more pleasant and similar to the input image while avoids affecting the decodability.

II. OVERVIEW

A standard QR code is composed of two-dimensional modules which are black and white squares containing bit information. The size of a QR code is controlled by the version number V . For a QR code of version V , the corresponding QR code size is $(17 + 4V) \times (17 + 4V)$ modules. Four error correction levels (i.e. L, M, Q and H) are used for determining the error correction capability of the QR code. The flow of how to synthesize a decodable QR code is briefly introduced as follows. The embedded message is first analyzed to determine the suitable version and the error correction level of the QR code. According to the associated encoding mode, the embedded message is encoded into a k -bits bit stream with the terminator symbols (0000) and padding bits. Reed-Solomon (RS) code is then applied to the k -bits bit stream for generating t parity bits used for detecting and correcting errors during scanning the QR code. Finally, the generated n -bits RS code (including m bits for *message bits*, p bits for *padding bits*, and t bits for *parity bits*) and the *function pattern* (i.e. the finder pattern, the alignment pattern, and the timing pattern used for improving the reading performance) are placed into the QR code with a masking operation [11]. As illustrated in Fig. 2(a), the blue/green/red regions indicate where the message/padding/parity bits are placed in the QR code, respectively. Each *message bit* is placed onto one module of the blue region in a vertical boustrophedon order starting at the bottom right corner, while each *padding/parity bit* is placed onto one module of the green/red region in the same manner.

QR code beautification is generally formulated as an optimization problem that minimizes the visual perception distortion subject to an acceptable decoding rate. However, since the

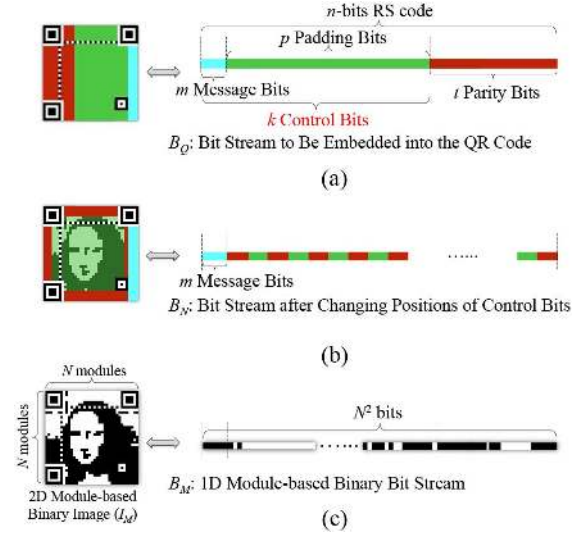


Fig. 2. (a) Illustration of how an RS decodable bit stream B_Q is placed onto a 2D QR code. (b) The new bit stream B_N obtained by changing positions of control bits and the corresponding 2D QR code. (c) The module-based binary image I_M and the corresponding 1D bit stream B_M .

arithmetic operations of RS code are defined over a finite field F , normal convex optimization approaches cannot be directly applied. Consequently, solving the optimization problem becomes a challenge, and related works usually take much time to generate an approximate result [9]. In this work, we propose a two-stage approach, in which a baseline QR code with robust decodability but poor visual quality is first synthesized based on an efficient Gauss-Jordan elimination procedure [10] and then a rendering mechanism is designed to improve the visual quality without seriously affecting the decodability of the QR code. Since both two stages can be done within 0.8 second, our method is efficient for real-time applications.

Fig. 3 shows the flowchart of our QR code beautifier. Given an input image I , we aim to generate a resultant QR code I_R , which is decodable and as similar as the visual content in I . That is, the color of a module m_i in I_R should be similar to the color of a pixel p_i in I . We first threshold the input image I in the L channel of the LAB color space and generate two binary images: one is the pixel-based binary result I_P obtained by thresholding each pixel independently according to a fixed threshold, and the other is the module-based binary result I_M acquired by assigning each module a binary value according to the averaging L value of all pixels inside the central region of the module. (The issue of how to determine the size of the central region will be discussed in Section IV.) The *function pattern* is then added to both of the binary images, and two intermediate binary images (i.e. I_P and I_M) are generated for the following processes.

For the module-based binary image I_M , we pre-mask it to keep the finding pattern recognizable and perform the QR code synthesis step to generate a baseline QR code image I_B . This QR code synthesis step is based on the method proposed by Russ Cox [10] and has been proved to have an acceptable decoding rate. We make some modification on the policy of selecting the *control bits* such that the region of interest (ROI) in I_M can be preserved in I_B . The baseline QR code image I_B can effectively

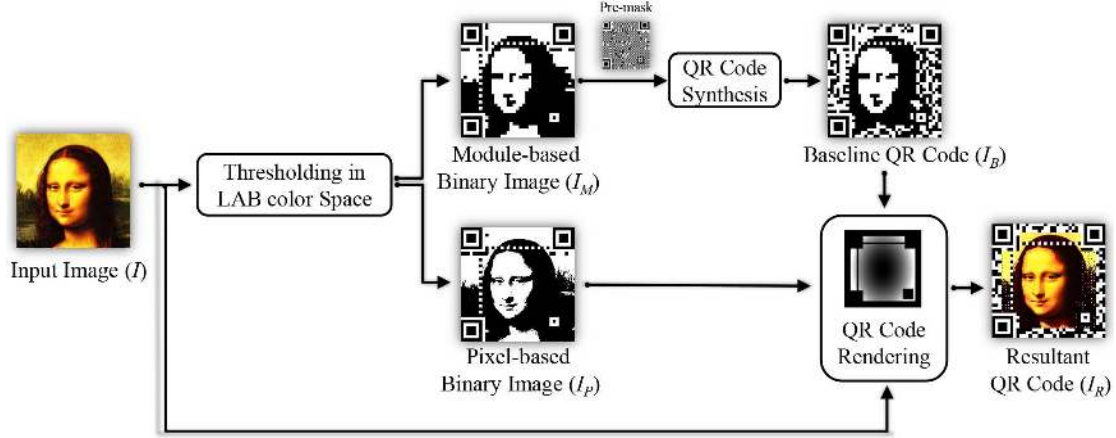


Fig. 3. Flowchart of the proposed QR code beautifier.

overcome the noise but is still a binary image with perceivable blocking artifacts compared to the original input image I . To further improve the visual quality of the QR code, we propose a rendering mechanism that takes the input image I , the pixel-based binary image I_P , and the baseline QR code image I_B into consideration to generate the final QR code result I_R . We will detail the QR code synthesis stage and the QR code rendering stage in the following sections.

III. BASELINE QR CODE SYNTHESIS

To tolerate the noises and increase the decoding capability of QR code, several pixels are grouped into one module to indicate 1-bit information. Therefore, the module-based binary image I_M is taken as our reference image and we hope our baseline QR code I_B preserves most important visual content of I_M while keeps its readability. We adopt Russ Cox's method [10] to generate I_B and make some modifications on control bit selection. Fig. 2 illustrates the main idea. Fig. 2(a) shows the QR code obtained by using the conventional way to place the RS decodable bit stream B_Q for the m -bits message. Fig. 2(c) shows the module-based binary image I_M and the corresponding 1D bit stream B_M . Our goal is to perform operations on bits of B_Q and transform it into a new bit stream B_N , as shown in Fig. 2(b), such that B_N is still RS decodable and similar to B_M . This goal can be achieved by the Gauss-Jordan Elimination procedure, which will be explained in the following section.

As shown in Fig. 2(a), an n -bits (n, k, t) RS code is composed of k control bits (including the encoded message, the terminator symbols, and the padding bits) followed by t parity bits for error correction. An important characteristic of RS code is that the XOR result of two RS code is also an RS code [12], which means we can build a valid RS code based on other two RS codes. For example, given the following three 5-bits RS codes with $k = 3$ and $t = 2$:

$$\begin{aligned} RS_1 &= \underline{100}10 \quad (K_1 = 100 \text{ and } T_1 = 10) \\ RS_2 &= \underline{010}11 \quad (K_2 = 010 \text{ and } T_2 = 11) \\ RS_3 &= \underline{001}01 \quad (K_3 = 001 \text{ and } T_3 = 01) \end{aligned}$$

$K_1, K_2,$ and K_3 form the basis set for the entire vector space of valid RS codes and the RS code B_i for $K_i=011, 110, 101,$

111, or 000 can be generated by applying XOR operations on the RS blocks of the basis set (i.e. $RS_1, RS_2,$ and RS_3). We call the first three bits the *control bits* since their values control the values of the *parity bits*. Among the basis set, there is only one basis having the 1 value on each *control bit*. If we want the second, third, and fourth bits to be the *control bits*, we have to replace RS_2 by $RS_1 \oplus RS_2$. The corresponding RS codes of the basis set for the new control bits are then updated as

$$\begin{aligned} RS'_1 &= \underline{100}10 \quad (K'_1 = 001 \text{ and } T'_1 = 10) \\ RS'_2 &= \underline{110}01 \quad (K'_2 = 100 \text{ and } T'_2 = 11) \\ RS'_3 &= \underline{001}01 \quad (K'_3 = 010 \text{ and } T'_3 = 01). \end{aligned}$$

The process of finding the RS codes for the new basis set $\{K'_1, K'_2, K'_3\}$ can be achieved by the Gauss-Jordan Elimination procedure. With $\{RS'_1, RS'_2, RS'_3\}$, we can further apply XOR operations to obtain the RS code for $K'_i = 011, 110, 101, 111,$ or 000 . Based on the above-mentioned characteristic of RS codes, we can formulate our problem as follows:

$$B_Q \xrightarrow{\Delta} B_N, \Rightarrow (n, k, t) \xrightarrow{\Delta} (n, k', t') \quad (1)$$

where B_Q is the original bit stream, B_N is the targeted bit stream, Δ is the Gauss-Jordan Elimination procedure, k' is the targeted control bits, and t' is the targeted parity bits. We can change the positions of *control bits* in the bit stream B_Q to generate a new bit stream B_N such that the *control bits* will be placed in the central part (ROI) of the 2D QR code [cf. Fig. 2(a) and (b)]. By applying the Gauss-Jordan Elimination procedure on the RS codes of the old basis set, we can find the RS codes of the new basis set. Moreover, we can set the values of the new *control bits* to be the same as the values in the ROI of I_M and obtain the corresponding RS code by applying the XOR operation on the RS codes of the new basis set. As a consequence, a decodable QR code I_B , which is similar to I_M , can be generated.

In Cox's method, the QR code is partitioned into salient region and non-salient region. As shown in Fig. 4(a), the salient (green) region is selected to be the control bits area. In contrast, we partition the QR code into two parts, i.e. border and central. As shown in Fig. 4(d), the green central area is selected to be the control bits area. After applying the Gauss-Jordan Elimination

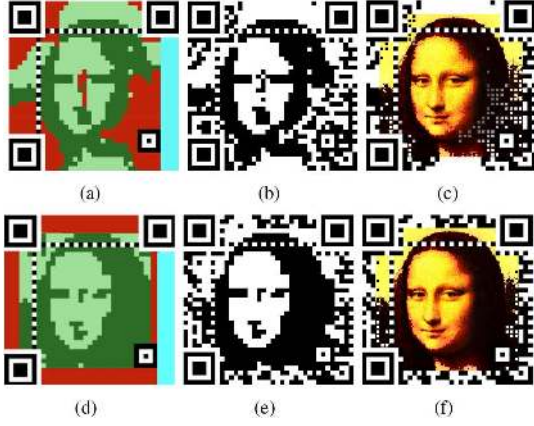


Fig. 4. Comparison of Cox's method and ours. (a) Cox's control bits selection strategy. (b) QR code generated based on (a) by applying the Gauss-Jordan Elimination procedure. (c) QR code generated based on (b) by applying our rendering mechanism introduced in Section IV. (d) Our control bits selection strategy. (e) QR code generated based on (d) by applying the Gauss-Jordan Elimination procedure. (f) QR code generated based on (e) by applying our rendering mechanism introduced in Section IV.

procedure based on (a) and (d) respectively, we found that our control bits selection strategy can produce much better result than Cox's [cf. Fig. 4(b) and (e)]. That is because parity bits are distributed over four border sides and visual content in the ROI of human perception can be well-preserved in this way. Moreover, if we further apply the proposed rendering mechanism introduced in Section IV to (b) and (e), our control bits selection strategy results in a decodable QR code with less visually-noisy pixels in the ROI [cf. Fig. 4(c) and (f)].

IV. QR CODE RENDERING

Although we can synthesize a decodable QR code as described in the previous section, the resultant QR code I_B is not satisfactory since it is a binary image with serious block artifacts [as shown in Fig. 4(e)]. To improve its visual quality, we further design a rendering process which also takes the original color image I and the pixel-based binary image I_P into consideration. As shown in Fig. 5, a specific module M in the resultant QR code I_R is composed of $N \times N$ pixels ($N = 9$ in our implementation). Chu *et al.* [8] had proved that if we want to correctly decode a module of size 3×3 (pixels), at least the center pixel (i.e., $1/3$ of the module size) should contain the correct information. Therefore, we divide each module into two parts, i.e. central part and non-central part, and apply different rendering strategy to each part. For pixels in the central part, the pixel values of the resultant QR code are rendered based on the baseline QR code I_B to ensure the decodability. For pixels in the non-central part, the pixel values of the resultant QR code are rendered based on the pixel-based thresholded image I_P to mitigate the block artifacts.

To be more precise, the brightness of each pixel (i, j) in a module of the resultant QR code is rendered by the following equation:

$$I_{i,j}^{I_R} = \begin{cases} (L_{i,j}^{I_B} - L_{i,j}^I) * \alpha + L_{i,j}^I, & (i, j) \text{ in central pixels;} \\ (L_{i,j}^{I_P} - L_{i,j}^I) * \alpha + L_{i,j}^I, & \text{otherwise} \end{cases} \quad (2)$$

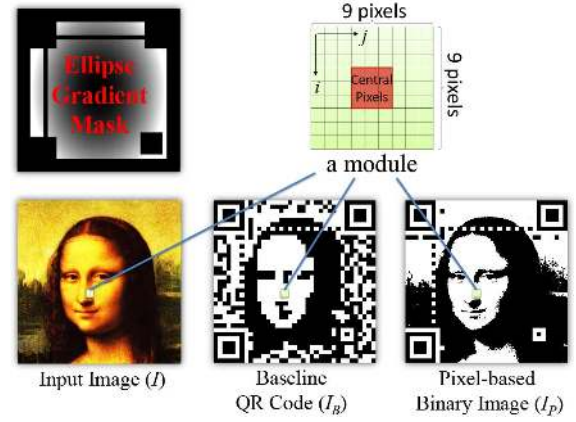


Fig. 5. Each module in the resultant QR code I_R is rendered by assigning values to pixels inside the module according to I , I_B , and I_P .

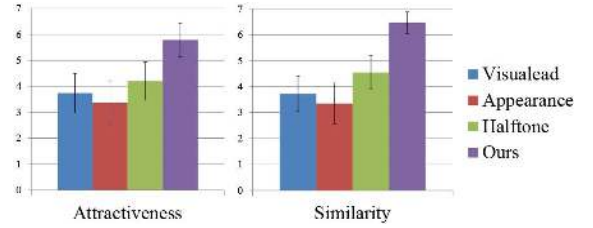


Fig. 6. Subjective test in terms of attractiveness and similarity with respect to the input image.

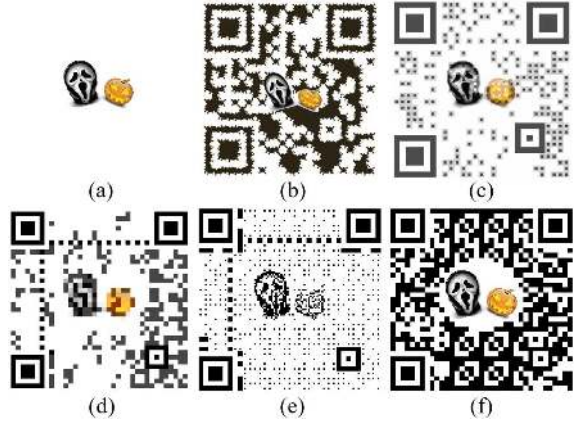
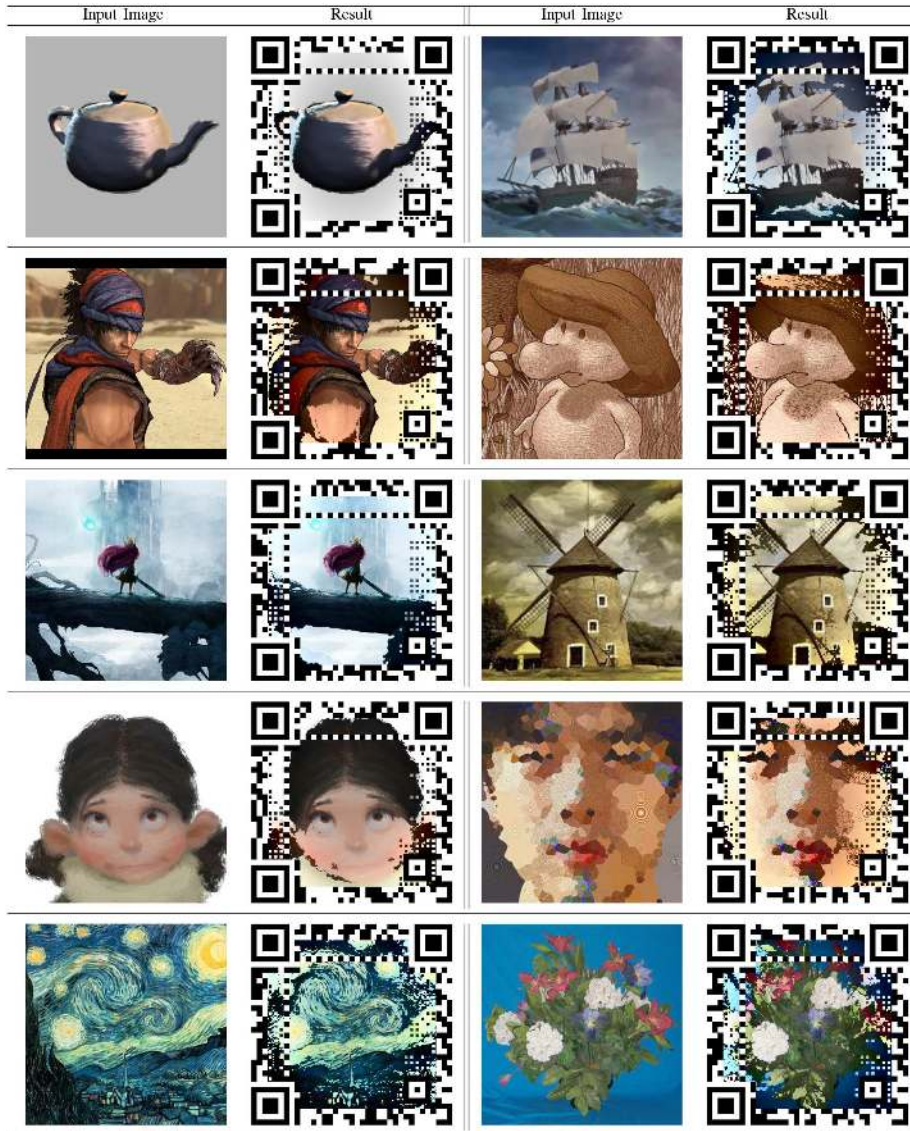


Fig. 7. (a) Input icon. (b)–(f) are the results generated by Artistic QR Code Embellishment [7], Visualead, Appearance-Based QR Code Beautifier [9], Halftone QR Codes [8], and our method, respectively.

where α is a parameter controlling the readability of the module m . When $\alpha = 0$, the resultant QR code is the same as the input image I but the readability is zero. When $\alpha = 1$, the central part of each module is the same as I_B , which is robust to noise while decoding. At the same time, non-central pixels are rendered the same as I_P , which preserves the edge details of the input image. We can simply use a constant α value for all pixels, but how to determine a proper constant value is also a problem. Instead of using a constant α for the whole QR code, we assign pixels with different α values and minimize the following cost function to

TABLE II
EXAMPLE RESULTS OF APPLYING THE PROPOSED QR CODE BEAUTIFICATION METHOD TO NPR IMAGES



automatically find the best $\alpha(x, y)$ for each pixel (x, y) in the resultant QR code I_R

$$\begin{aligned}
 E = \sum_{x,y} \left\{ \delta(x, y) \times \|L_{x,y}^{I_R} - L_{x,y}^{I_B}\|^2 \right. \\
 + (1 - \delta(x, y)) \times \|L_{x,y}^{I_R} - L_{x,y}^I\|^2 \\
 \left. + \|\alpha(x, y) - G(x, y)\|^2 \right\} \quad (3)
 \end{aligned}$$

where

$$\begin{aligned}
 G(x, y) = \\
 \min \left\{ \rho_x \times \left(\frac{2 \times |x - c_x|}{I_w} \right)^2 + \rho_y \times \left(\frac{2 \times |y - c_y|}{I_h} \right)^2, 1 \right\}. \quad (4)
 \end{aligned}$$

$\delta(x, y)$ is an indicator function and $\delta(x, y) = 1$ when (x, y) is a central pixel; otherwise $\delta(x, y) = 0$. The first term in the summation ensures that each central pixel in the resultant QR code I_R should be similar to the baseline QR code I_B such that the resultant QR code can be decodable. The second term in the summation ensures that each non-central pixel should be similar to the original image I . The final term in the summation constrains $\alpha(x, y)$ to be similar to an ellipse gradient function as illustrated in Fig. 5, where ρ_x and ρ_y are set to be 1 and 1.5, respectively; c_x and c_y are the center positions of the QR code's width (I_w) and height (I_h), respectively.

ρ_x and ρ_y control the values in the ellipse gradient mask and can be determined according to the size of padding-bit region, which is affected by two factors, i.e. the embedded message length and the error correction level. Therefore, instead of using fixed values, ρ_x and ρ_y can be adjusted by

$$\rho_x = \frac{\beta \times (m + t)}{n} \quad \text{and} \quad \rho_y = \frac{\gamma \times (m + t)}{n} \quad (5)$$

TABLE III
DISSIMILARITY BETWEEN THE INPUT IMAGE AND THE QR CODES
GENERATED BY DIFFERENT METHODS

Image #	[8]	[10]	[11]	Ours
Image 1	63.20	147.23	67.45	41.25
Image 2	64.55	171.32	67.26	34.76
Image 3	61.19	127.94	77.80	47.17
Image 4	62.85	177.80	68.91	46.88
Image 5	62.78	214.12	73.09	43.17
Image 6	63.63	203.54	63.64	51.37
Image 7	70.89	160.99	65.48	28.54
Average	64.16	171.85	69.09	41.88

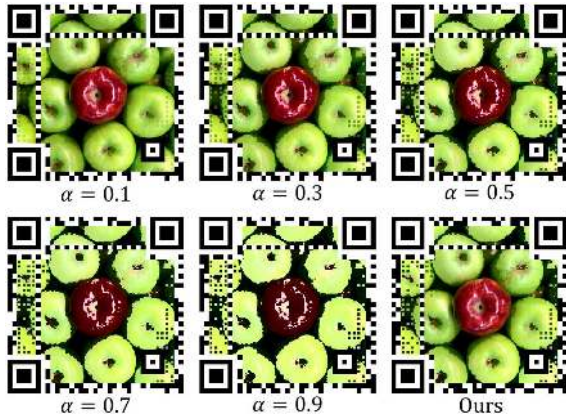


Fig. 8. Comparison of using different α values in the QR code rendering step.

TABLE IV
DECODING RATES ON DIFFERENT MOBILE DEVICES

Mobile Phone	App	Success Rate
Sony ZL	QR Droid	100%
	scanner pro qr code reader	100%
	QR BARCODE SCANNER	100%
	QR Code Reader	100%
	Qrcode Scanner	100%
	Neo Reader	94%
HTC butterfly	QR Droid	100%
	scanner pro qr code reader	100%
	QR BARCODE SCANNER	100%
	QR Code Reader	100%
	Qrcode Scanner	100%
	Neo Reader	100%
Redmi	QR Droid	100%
	scanner pro qr code reader	100%
	QR BARCODE SCANNER	100%
	QR Code Reader	100%
	Qrcode Scanner	100%
	Neo Reader	100%

where $\beta = 2.5$, $\gamma = 3.75$, m is the number of message bits, t is the number of parity bits, and n is the total number of bits in an RS code. The ellipse gradient mask ensures that modules nearby the image border would have larger α values (i.e. have higher readability) and modules nearby the center region would have smaller α values (i.e. preserve most visual content of the input image). Note that the minimization of the cost function is subject to $0 \leq \alpha(x, y) \leq 1$ and $\sum_{x,y} \alpha(x, y) \geq \varphi$, where φ is a constant. To increase decoding rate of the QR code, $\sum_{x,y} \alpha(x, y)$ must be larger than $\sum_{x,y} G(x, y)$. Constrained linear least-squares is then applied to solve the minimization problem.

TABLE V
THE ERROR CORRECTION CAPABILITY OF THE BEAUTIFIED QR CODES

Our Result #	Errors (bytes)	Error Rates
Result 1	4	3.0%
Result 2	1	0.7%
Result 3	0	0%
Result 4	3	2.2%
Result 5	7	5.2%
Result 6	6	4.5%
Result 7	3	2.2%
Result 8	5	3.7%
Result 9	3	2.2%
Result 10	3	2.2%
Result 11	3	2.2%
Result 12	2	1.5%
Result 13	10	7.5%
Result 14	6	4.5%
Result 15	5	3.7%
Result 16	2	1.5%
Result 17	5	3.7%
Average	4	3.0%



Fig. 9. QR codes generated by our method with different version number ($V = 11, 9, 6, \text{ and } 3$).



Fig. 10. QR codes generated by our method with images of different sizes.

V. RESULTS AND DISCUSSION

Visual quality and time complexity: To evaluate the performance of our method, we invited 42 people (including 16 male and 26 female) to conduct the subjective test. A variety of images were taken as the input images for embedding the source text “ieeexplore.ieee.org” based on four different methods, i.e. Visualead, Appearance-Based QR Code Beautifier [9], Halftone QR Codes [8] and our method. The produced QR codes corresponding to each method are presented in Table I, where the version of the QR code is set to be 5 and the error correction level L is used. Each individual was asked to answer a qualitative questionnaire by using 1 to 7 (1 means less satisfactory and 7 means excellent) to score the attractiveness of the resultant QR codes and the similarity between the resultant QR code and the original input image. Our method outperforms others in all cases and Fig. 6 shows the average/variance of the scores for each method. Since Lin’s approach [7] can only place small icons into the QR code, we further compare with their work by first resizing the input image into a small icon and then embed the message into the QR code. Fig. 7 shows that Lin’s method results in shape distortion of the original icon while our method preserve both of the shape and the color of the original icon. We also applied our method to NPR images (some examples are shown in Table II), and the results



Fig. 11. QR code results rendered by assigning ρ_x and ρ_y with different values, respectively.

demonstrate that our method is general to be applied to all kinds of images. Moreover, our method takes less than a second to generate the resultant QR code on a PC with Intel(R) Core i7-3770 CPU (3.4 GHz), which is computationally efficient.

In addition to subjective user test, we also objectively evaluated the dissimilarity between the generated QR code and the input image by simply summing the RGB-based color distance of all pixels in the ROI. Table III lists the dissimilarity between the seven input images in Table I and the corresponding QR codes generated by different methods. The results objectively prove that our method can generate a decodable QR code which is more similar to the input image compared to the state-of-the-art methods.

The influence of α value: As mentioned in Section IV, using a larger α value to render each module will lead to better decoding capability, while applying a smaller α value can keep the resultant QR code I_R more similar to the original image I . Fig. 8 illustrates this phenomena, where the QR code with $\alpha = 0.1$ is undecodable and the one with $\alpha = 0.9$ has poor visual quality. Instead of using a constant α value for all modules, our method assigns larger α values to modules near the image border and smaller α values to modules in the center region of the image. As shown in Fig. 8, our method successfully generates a decodable QR code and simultaneously keeps most visual content of the input image.

Correctness of QR code decoding: We evaluated the correctness of decoded messages on three different mobile devices (Sony ZL, HTC butterfly, Redmi) with six various QR code decoders (QR Droid, scanner pro qr code reader, QR BARCODE SCANNER, QR Code Reader, Qrcode Scanner, Neo Reader). 17 images (our results in Table I and II) are used to generate the beautified QR codes based on our method and the decoding rates are reported in Table IV. Only one fail occurred when we used Sony ZL and Neo Reader to scan the generated QR code. The other two mobile devices can successfully decode our QR codes with 100% decoding rate. The loss of decoding might result from the poor camera quality of the mobile device or improper α values.

In addition to the above descriptions, we also applied ZXing (Zebra Crossing) decoder³ to evaluate the errors (bytes or codewords) and error rates of our beautified QR codes (our results in Table I and II). Table V shows the error correction capability of our beautified QR codes. In our experiment, the beautified QR codes are generated by using version 5 and error correction

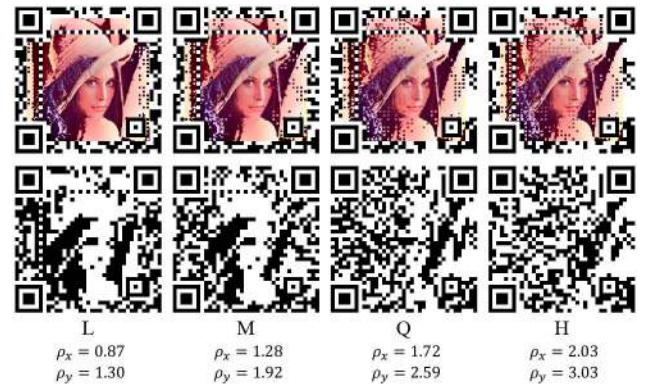


Fig. 12. QR-code results generated by using various error correction levels, including L, M, Q, H. Top and bottom rows represent the beautified QR code results and the baseline QR codes, respectively.

level L. That is, the total length of the QR code would be 134 bytes, the control codeword length would be 108 bytes, and the error correction codeword length would be 26 bytes. For each QR code, we use ZXing decoder to decode the information and the error rate is computed by $(\frac{\text{Error bytes}}{\text{Total bytes}} \times 100\%)$. The permitted error rate of the generated QR code with version 5 and error correction level L can be approximated as $(26/2)/134 = 0.097(9.7\%)[13]$. As shown in Table V, the error rate of each beautified QR code is less than 9.7% and the average error rate is about 3.0%, which means our beautified QR code is robust to be decoded correctly.

The influence of image size and QR code size: In addition to visual quality, another well known problem of QR codes is how to determine the proper size for a readable QR code. The QR code size is determined by the version number V . We generate QR codes of different sizes using the same input image as shown in Fig. 9 ($V = 11, 9, 6,$ and 3). The results generated by our method are decodable as long as the DPI is sufficient to clearly see the modules. Due to different DPI settings of various displays or printers, it is suggested that the QR code images in our paper to be zoomed in before being scanned. Clear submodules would result in better readability of the QR code. We also resize the input image into different sizes and embed them into the QR code, as shown in Fig. 10. All these four QR codes are decodable and the visual content are preserved successfully.

The influence of ρ_x and ρ_y : In the rendering step, the values in the ellipse gradient mask are determined by ρ_x and ρ_y , which can be fixed values or can be adjusted according to the embedded message length and the error correction level [as formulated in (5)]. We further conducted experiment to investigate the influence of ρ_x and ρ_y . Fig. 11 shows the results of assigning

³“Multi-format 1D/2D barcode image processing library implemented in Java, with ports to other languages,” [Online]. Available: <https://github.com/zxing/zxing>



Fig. 13. QR code results generated by embedding messages of different lengths. (a)–(e) are the results of embedding 10, 20, 40, 80, and 154 alphanumeric symbols, respectively. (f) The result of embedding a shortened URL for a message with 154 alphanumeric symbols. Top and bottom rows represent the beautified QR code results and the baseline QR codes, respectively.

ρ_x and ρ_y with different values (in this case we use error correction level L). Smaller values keep most content of the original image and make the QR code more pleasant.

The influence of error correction level: The error correction level controls the proportion of parity bits. When we encode messages of the same length, the proportion of parity bits in level L would be less than that in level H. That is, the region for control bits would be larger in level L than that in level H. Fig. 12 shows that when error correction level is lower, we would have less parity bits and the image quality of our beautified QR code would be better. Thus, we utilize error correction level L to produce the best result. Even though we apply level H to generate the result, we can still preserve most of the image content and the result is also decodable.

The influence of message length: The message length controls the proportion of padding bits and also affects our QR code result. In Fig. 13, the error correction level is fixed as L and messages of different lengths are embedded for comparison. As the length of the embedded messages gets longer, the proportion of padding bits would decrease and consequently results in poor visual quality, as shown in Fig. 13(a)–(e). Fortunately, this problem can be solved by using an URL shortener [see Fig. 13(f)], as used in the Visualead method. The URL shortener (ie. Google URL shortener) can create a shorter URL for the original input URL, so that the affection of embedding long messages can be reduced while generating the beautified QR code. The user will be firstly connected to this shorter URL and then be redirected to the original URL or message. Note that the values of ρ_x and ρ_y for the ellipse gradient mask are automatically adjusted according to the embedded message length and the error correction level as described in Section IV.

The influence of lighting condition and viewing angle: For each beautified QR codes generated by our method (30 beautified QR codes), we simulated images captured under different lightings by adding an intensity value l ($l = -30, -20, -10, 0, +10, +20, +30$) to all pixels in the QR code image. We also simulated images taken by different camera viewing orientations by rotating the generated QR code image with an out-of-plane rotation angle r ($r = 10^\circ, 20^\circ, 30^\circ, 40^\circ, 50^\circ, 60^\circ, 75^\circ$). We then used a mobile device (HTC butterfly) equipped with six various QR code de-

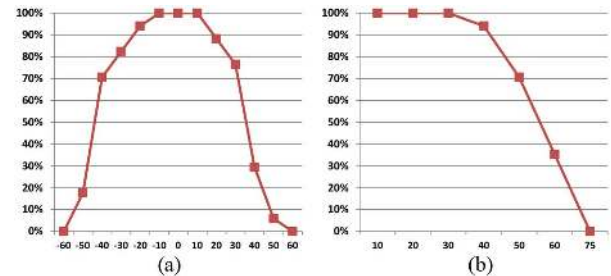


Fig. 14. (a) Success rates of decoding the beautified QR code results under different lighting conditions. (b) Success rates of decoding the beautified QR code results with different camera viewing angles.

coders (QR Droid, scanner pro qr code reader, QR BARCODE SCANNER, QR Code Reader, Qrcode Scanner, Neo Reader) to decode all the bits in the simulated images. Fig. 14 shows the success rates of decoding with respect to different lighting conditions and camera viewing angles. We can ensure the success of decoding in indoor that has a sufficient light. Unfortunately, we cannot ensure successful decoding in dark or under sunlight. Moreover, even though the viewing angle is rotated by 30° , the generated QR codes can be correctly decoded.

Application: The nature of our rendering mechanism makes the resultant QR code superior to existing QR codes in term of visually quality. We further apply it to synthesize frame-by-frame QR codes in video clips. Although the temporal coherence is not taken into consideration, the generated results are much more satisfactory than other methods because the visual content in the center part remains almost the same as the original video and the flickering effect only occurs around the frame boundary. The readability of the QR code is also robust as long as the PPI (pixel per inch) is large enough.

Limitations: Our method works for all kinds of images. However, for images whose pixel brightness values distribute around 128, the thresholded images I_M and I_P might seriously lose edge information of the input image I . In this case, we have to first enhance the image contrast using the mapping function as shown in Fig. 15(a) such that the generated QR code could have better visual quality. Fig. 15(b) and (c) are the QR codes generated by our method with contrast enhancement in LAB and RGB color spaces, respectively. We observe that conducting contrast

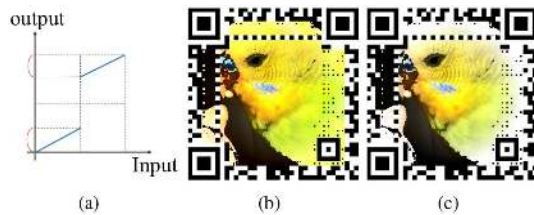


Fig. 15. (a) Mapping function for contrast enhancement. (b) QR code generated by our method with contrast enhancement in LAB color space. (c) QR code generated by our method with contrast enhancement in RGB color space.

enhancement in LAB color space would result in better visual quality than in RGB color space for most of the low-contrast images.

VI. CONCLUSION

In this work, we propose an efficient two-stage approach to generate QR codes with high quality visual content. A decodable baseline QR code with poor visual quality is first synthesized based on the Gauss-Jordan elimination procedure and then a rendering mechanism is designed to improve the visual quality while avoid affecting the decodability of the QR code. The experimental results show that the proposed method outperforms the existing works considering the appearance of the QR code and the processing complexity is near real-time.

ACKNOWLEDGMENT

The authors would like to thank Dr. Y.-H. Lin, Dr. J.-L. Wu, and Dr. H.-K. Chu for providing the source codes of their works, so the authors could easily compare their method with the state-of-the-art works. They also appreciate Dr. S.-J. Luo's help on providing example results of their work for evaluation.

REFERENCES

- [1] T.-W. Kan, C.-H. Teng, and W.-S. Chou, "Applying QR code in augmented reality applications," in *Proc. 8th Int. Conf. Virtual Reality Continuum Appl. Ind.*, 2009, pp. 253–257.
- [2] T. Anezaki, K. Eimon, S. Tansuriyavong, and Y. Yagi, "Development of a human-tracking robot using QR code recognition," in *Proc. 17th Korea-Japan Joint Workshop Frontiers Comput. Vis.*, 2011, pp. 1–6.
- [3] B. Erol, J. Graham, J. J. Hull, and P. E. Hart, "A modern day video flip-book: Creating a printable representation from time-based media," in *Proc. 15th Int. Conf. Multimedia*, 2007, pp. 819–822.
- [4] D. Haisler and P. Tate, "Physical hyperlinks for citizen interaction," in *Proc. Int. Conf. Multimedia*, 2010, pp. 1529–1530.
- [5] T. Nikolaos and T. Kiyoshi, "QR-code calibration for mobile augmented reality applications: Linking a unique physical location to the digital world," in *Proc. ACM SIGGRAPH*, 2010, pp. 144:1–144:1.
- [6] A. Falcon, "40 gorgeous QR code artworks that rock," 2013 [Online]. Available: <http://www.hongkiat.com/blog/qr-code-artworks/>, Accessed on: Dec. 2013
- [7] Y.-S. Lin, S.-J. Luo, and B.-Y. Chen, "Artistic QR code embellishment," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 137–146, 2013.
- [8] H.-K. Chu, C.-S. Chang, R.-R. Lee, and N. J. Mitra, "Halftone QR codes," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 217:1–217:8, 2013.
- [9] Y.-H. Lin, Y.-P. Chang, and J.-L. Wu, "Appearance-based qr code beautifier," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 2198–2207, Dec. 2013.

- [10] R. Cox, "Qartcodes," Apr. 2012 [Online]. Available: <http://research.swtch.com/qart>, Accessed on: Oct. 2012
- [11] *Information Technology—Automatic Identification and Data Capture Techniques—QR Code 2005 Bar Code Symbology Specification*, ISO/IEC 18004:2006, 2006.
- [12] R. Cox, "Finite field arithmetic and Reed-Solomon coding," Apr. 2012 [Online]. Available: <http://research.swtch.com/field>, Accessed on: Oct. 2012
- [13] H. Kato, K. T. Tan, and D. Chai, *Barcodes for Mobile Devices*. Cambridge, U.K.: Cambridge Univ. Press, 2010.



graphics.

Shih-Syun Lin received the B.S. degree in applied mathematics from Providence University, Taichung, Taiwan, in 2007, the M.S. degree from the Graduate Institute of Educational Measurement and Statistics, National Taichung University, Taichung, Taiwan, in 2010, and is currently working toward the Ph.D. degree in computer science and information engineering at National Cheng Kung University, Tainan, Taiwan.

His research interests include video retargeting, mesh deformation, pattern recognition, and computer



Min-Chun Hu (S'07–M'11) received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2004 and 2006, respectively, and the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan, in 2011.

She was a Post-Doctoral Research Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan, from 2011 to 2012. She is currently an Assistant Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. Her research interests include digital signal processing, digital content analysis, pattern recognition, computer vision, and multimedia information system.



Chien-Han Lee received the B.S. degree in information and computer engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2012, and the M.S. degree in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2014.

He is currently with MACHVISION Inc., Hsinchu, Taiwan. His research interest include computer graphics.



Tong-Yee Lee (M'04–SM'10) received the Ph.D. degree in computer engineering from Washington State University, Pullman, WA, USA, in 1995.

He is currently a Chair Professor with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan. He also leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University. His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and

media resizing.

Dr. Lee is a member of the ACM.