Halftone QR Codes

Hung-Kuo Chu[1] Chia-Sheng Chang[1] Ruen-Rone Lee[1] Niloy J. Mitra[2]

[1]National Tsing Hua University [2]University College London

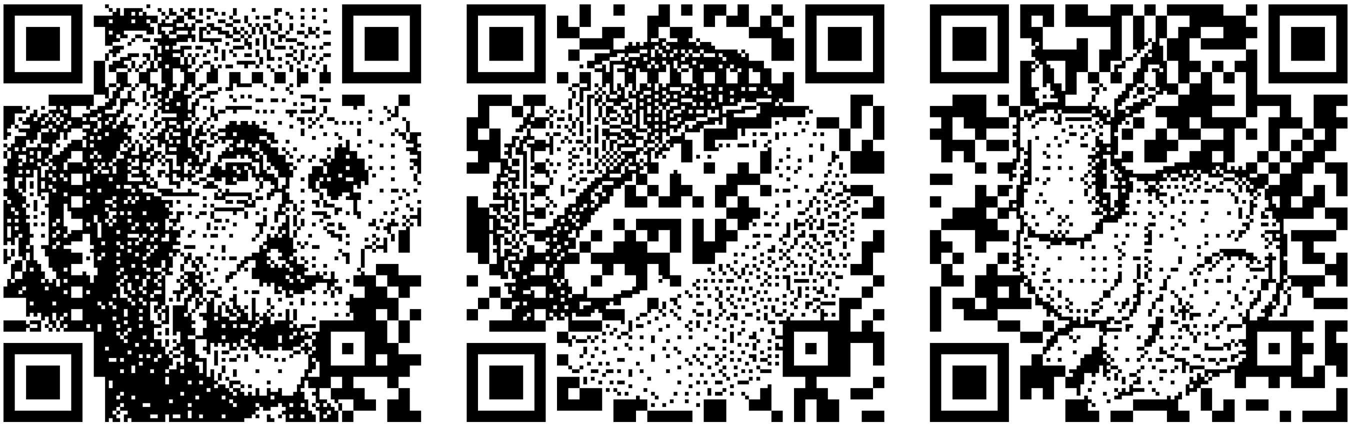


Figure 1: *Three halftone QR codes generated by our method. By using a new representation model that minimally binds to the appearance of QR code, our approach is able to combine halftone images with ordinary QR codes without compromising its readability.*

Abstract

QR code is a popular form of barcode pattern that is ubiquitously used to tag information to products or for linking advertisements. While, on one hand, it is essential to keep the patterns machine-readable; on the other hand, even small changes to the patterns can easily render them unreadable. Hence, in absence of any computational support, such QR codes appear as random collections of black/white modules, and are often visually unpleasant. We propose an approach to produce high quality visual QR codes, which we call *halftone QR codes*, that are still machine-readable. First, we build a pattern readability function wherein we learn a probability distribution of what modules can be replaced by which other modules. Then, given a text tag, we express the input image in terms of the learned dictionary to encode the source text. We demonstrate that our approach produces high quality results on a range of inputs and under different distortion effects.

CR Categories: I.4.9 [Image Processing and Computer Vision]: Applications;

Keywords: Non-Photorealistic Rendering, Halftone, QR code

Links: DL PDF Web Video Code

ACM Reference Format
Chu, H., Chang, C., Lee, R., Mitra, N. 2013. Halftone QR Codes. ACM Trans. Graph. 32, 6, Article 217 (November 2013), 8 pages. DOI = 10.1145/2508363.2508408 http://doi.acm.org/10.1145/2508363.2508408.

1 Introduction

Quick Response Code, abbreviated as QR code®, is a two-dimensional matrix encoding consisting of black and white squares, called *modules*, forming a machine-readable barcode to tag information onto products. Originally designed by Denso Wave for the automotive industry, QR code has quickly been adapted as a fast and effective way to embed digital content and is extensively used in diverse fields including manufacturing, marketing, etc. While being an excellent machine readable format, visually QR code remains a clutter of black and white squares that can easily disrupt the aesthetic appeal of its parent product.

Since QR codes often take up a non-negligible display area, there is a growing demand for producing visually appealing QR codes. Such codes that incorporate high-level visual features such as colors, letters, illustrations, or logos are referred to as *visual QR codes*. However, creating a visually interesting QR code without compromising its readability is non-trivial. The key challenge arises due to the lack of proper understanding or analytical formulations capturing the stability (i.e., validity) of QR codes under variations in lighting, camera specifications, and even perturbations to the QR codes [DENSO WAVE 2003; Winter 2011]. Patented and ill-documented algorithms employed for reading QR codes cause further difficulties. Consequently, existing approaches are mostly ad hoc and often end up favoring readability at the cost of sacrificing visual quality.

A common strategy to generate visual QR codes relies on inbuilt error correcting capabilities of QR codes to restore from missing or corrupted modules (see Figure 2(a)). In absence of suitable analytical or computational support, such approaches involve tedious trial-and-error runs to produce visual QR codes with little or no control over the final quality. As a result, the resolution and quality of the results are strongly dependent on and restricted by the settings used to generate the QR codes. Another heuristic is to modify a module's appearance while keeping its concentric region untouched, and uniformly blending the neighboring regions with the code modules (see Figure 2(c)). However, due to the tight binding to the appearance of QR code, such blending-based approaches

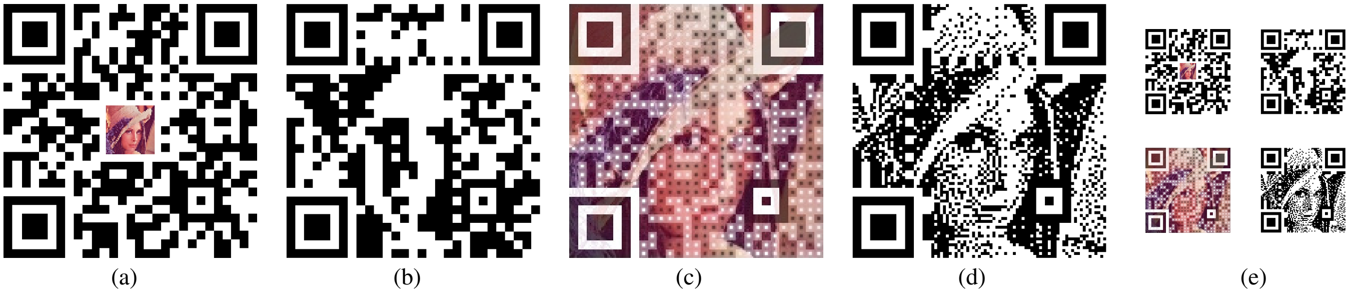


Figure 2: *Various visual QR codes generated by: (a) Unitag [Laporte 2012], (b) QArt [Cox 2012], (c) Visualead [Peled 2012], and (d) our method. (e) our result shows even better visual quality of embedded image as viewed from a distance (using 3X zoom out in this case).*

typically suffer from artifacts like corruption of salient image features and large variations in output quality for the same image with different QR codes (see supplementary video). Such methods offer little user control and at best produce mediocre results.

In this paper, we present an automatic algorithm to create a new type of visual QR code, called *halftone QR code*, that combines halftone images with QR codes (see Figure 1 and Figure 2(d)). To eliminate the aforementioned problems, the key insight is a representation model that minimally binds to original modules and is flexible to adapt to target halftone images. In our approach, we subdivide each module into 3×3 submodules and bind the module's color to the center submodule, which leaves the remaining eight submodules free to independently change their appearance. We use this flexibility to characterize the appearance of modules using a set of *binary patterns*. In addition, we evaluate the *pattern reliability* of each binary pattern that represents a probability of retaining module's readability when it is replaced by the pattern. Specifically, we estimate the probability by learning from a database of synthetic QR codes that is automatically generated using binary patterns and a barcode reader is employed to statistically evaluate pattern's reliability. Finally, to generate halftone QR codes, we introduce a novel optimization method to compute and assign a binary pattern to each module by balancing two competing terms, *reliability* and *regularization*, corresponding to preferring the assignment of high reliability patterns and regularizing the appearance of modules towards target halftone images, respectively. (Please refer to the project page to create your own halftone QR codes for non-commercial use.)

Although our algorithm runs automatically, we also provide tools to locally adjust the visual quality of the resultant QR codes. We extensively tested our method on a variety of inputs and also manually scanned the resultant QR codes using a smartphone with popular barcode readers to validate the reliability and the readability of the generated halftone QR codes.

Contributions. Our main contributions include:

- proposing a new type of visual QR code, *halftone QR code*, and an automatic algorithm to produce such codes at controllable level of readability;
- encoding module's appearance using a set of binary patterns that are minimally bound to modules and flexible to adapt to halftone images;
- characterizing pattern reliability that is evaluated by an automatic procedure, which computes statistics of a barcode reader reading a database of synthetic QR codes; and
- a pattern assignment optimization method that maximizes the readability based on pattern reliability and regularizes the module appearance via halftone image.

2 Related Work

The widespread use of QR codes motivates a vast creation of eye-catching visual QR codes. Most of the visual QR codes are creative artworks made by skilled artists or design experts. There is a web site, QR code®Artist [D'Haem 2010], that keeps track of news and related resources about creative QR code artworks, and a plentiful art gallery could be found in a flickr QR code art group [flickr 2009]. Owing to the commercial value behind such visually appealing QR codes, most resources for creating artistic QR codes are proprietary and only a few online tutorials [Chan 2011; Scheib 2010] are available. Commonly used strategies by artists to customize a QR code include adding colors to modules, rounding the hard edges, and embedding a small image in a QR code that relies on inbuilt error correction capacity of QR code to restore from missing modules (see Figure 2(a)). These attempts generally require a trial-and-error style process to manually refine their works followed by a readability verification process as evidenced in online custom QR code design tools [Laporte 2012].

Cox [2012] proposed a complicated algorithm to embed a binary image into a QR code during the data encoding stage of generating the code. He carefully investigated the internal structure of QR code and the logic behind data encoding, and designed an algorithm to encode image content as redundant numeric strings appended to the original data as shown in Figure 2(b). However, this technique works only for URL type data string and the quality of embedded image is limited by the length of encoded URL. Similarly, Duda [**?**] attempted to embed halftone images into QR codes using a constrained coding technique. While a sophisticated analysis of freedom bits is proposed for embedding image pixels, it currently works for images of monotonic shapes and produces mediocre results. An application of extending to delicate halftone images is not addressed.

Our work bears some resemblance to two proprietary techniques developed by Visualead [Peled 2012] and LogoQ [A T Communications Co. 2007]. Both approaches adopt a similar idea to automatically blend the input image with the QR code modules. Specifically, they keep a concentric region of modules untouched and uniformly blend the neighboring regions with image content in a manner that preserves the original contrast between modules (see Figure 2(c)). While their approaches produce colorful and attractive visual QR codes, the tight binding between the image and appearance of QR code causes the artifacts like corruption at salient image features and large variations in output quality for the same image with different QR codes (see supplementary video). All the existing approaches offer little user control, and the quality of visual QR codes is strongly dependent on either the settings or the appearance of QR codes. Moreover, the visual quality and clarity of their results drop dramatically as viewed from a distance (see Figure 2(e)). In this paper, we propose an automatic algorithm to generate halftone

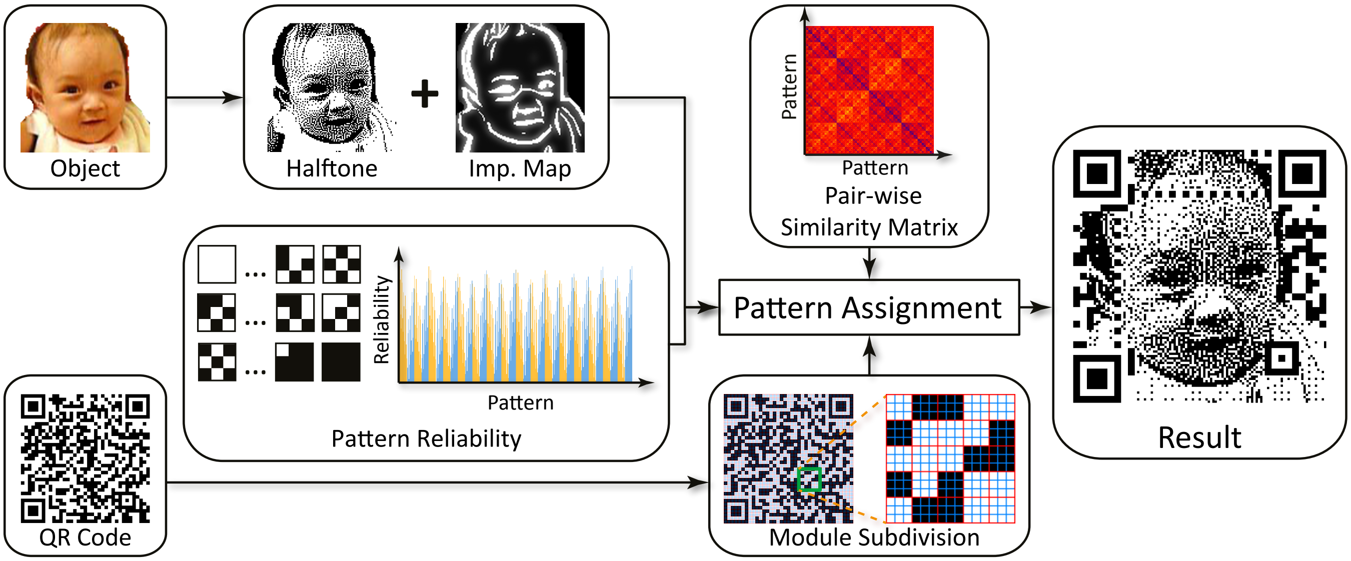


Figure 3: *System overview.*

QR codes at controllable level of readability. Our approach largely looses the binding between the QR code and halftone image, thus results in high quality visual QR codes which are resistant to the variation of QR codes.

Computer aided design of recreational arts has been extensively studied in various contexts including emerging images [**?**], shadow art [Mitra and Pauly 2009], camouflage images [Chu et al. 2010], pixel art [Kopf and Lischinski 2011], etc. Further, collage art [Gal et al. 2007; Huang et al. 2011] and text art [Xu et al. 2010; Maharik et al. 2011] that aim at optimizing the assemblage to retain local element characteristics while approximating a global shape, are relevant to our work. However, instead of maintaining shape and identifiability of local elements, our goal is to retain readability of the QR codes. This is a much stricter constraint.

3 Overview

Figure 3 provides a workflow of our halftone QR code synthesis. First, we generate a QR code using a data encoding library [Kentaro 2006] that adheres to ISO standard [ISO/IEC 18004 2006]. The setting used to generate a QR code consists of three parameters: data string, symbol version*, and error correction level. While the data string represents the encoded information that determines the appearance of QR code, symbol version and error correction level control the size and error correction capacity of QR code, respectively. The QR code modules are black or white squares, and can be classified into two categories according to their functionality. The first category contains data modules, which are modules that represent input data or error correction codes; while, the second category are modules that are used to improve the reading performance (e.g., for alignment, rectification, etc.). Because the readability of QR code is very sensitive to the correctness of modules in the second category, our algorithm leaves these modules untouched and only manipulates the data modules. We recommend readers to refer to the QR code ISO standard [ISO/IEC 18004 2006] and the official web site [DENSO WAVE 2003] for technical details regarding QR code generation.

A key component in generating halftone QR code is a representation model that minimally binds to the original module, and yet is flexible to adapt target halftone image. To this end, we propose a model that subdivides each module into 3×3 submodules and binds module's color to the center submodule, while leaving remaining eight submodules free to independently change their appearance. By rendering a subdivided module as a binary image of 3×3 pixels, we obtain a set of 512 *binary patterns* (or simply *patterns* hereafter), which is used to characterize the appearance of modules. For each pattern, we introduce a novel term *pattern reliability*, which is modeled as a probability that a module's readability is not compromised by a pattern substitution, and is evaluated through a large database of synthetic QR codes. Note that in absence of any analytic function quantifying readability of QR code under variations, pattern reliability characterizes the preference among patterns towards controllable levels of readability.

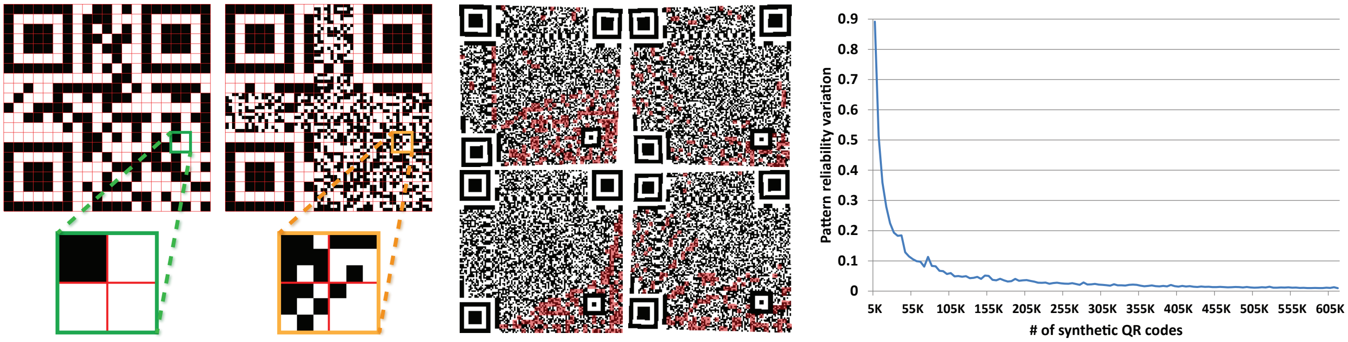
We formulate the synthesis of halftone QR codes as an optimization that assigns patterns to modules by computing an objective function comprising two terms, *reliability* and *regularization*. While reliability prefers selection of high reliability patterns in order to maximize the readability of modules; regularization aims to control the appearance of modules via the target halftone image using a similarity distance metric. We introduce a single control parameter to balance these two terms and solve the pattern assignment using a graph cut formulation.

4 Algorithm

Given an object image and its desired size, position, and orientation in the QR code, our system automatically converts the object image to a halftone image I_f using the method of Chang et al. [2009], and applies an image filter [Kyprianidis and Döllner 2008] to generate an importance map I_m of object image (see Figure 3). The importance map highlights salient image features and is later used to guide the QR code generation.

Pattern assignment optimization. We denote those data modules covered by the object image as $M = \{m_i = (I_i^m, c_i^m, w_i)|i = 1, \ldots n\}$ where, I_i^m is a local 3×3 image patch from I_f; c_i^m is the color of module; w_i represents the importance weight calculated

*There are 40 symbol versions, from 21×21 modules (version 1) to 177×177 modules (version 40) with each higher version adds additional 4 modules per side.



(a) (b) (c)

Figure 4: *(a) A synthetic QR code (right) generated from a QR code (left) using random pattern assignment. (b) Examples of synthetic QR codes with random perturbation. Red highlights indicate modules that are incorrectly decoded by ZBar [Brown 2007]. (c) The variation of estimated pattern reliability converges near the size of 0.6 million synthetic QR codes.*

by averaging the values from I_m. The set of patterns is denoted as $P = \{p_i = (I^p_i, c^p_i, r_i) | i = 1, \ldots 512\}$ where, I^p_i is a 3×3 binary image with color c^p_i at center pixel, and r_i represents pattern reliability in the range [0, 1.0] with higher values denoting higher reliability. We later discuss how to evaluate such pattern reliability.

Our goal is to assign each module a pattern picked from P such that two energy terms, reliability and regularization, are balanced. Let the patterns assigned to the modules be $P' = \{p'_i = (I^p_{f(i)}, c^p_{f(i)}, r_{f(i)}) | i = 1, \ldots n\}$ where, $f(i)$ represents index of a pattern in P that is assigned to module m_i.

The *reliability energy* prefers the assignment of patterns with higher reliability in order to maximize the readability of modules. As modules may cover different image regions with different saliency, we account for the modulation based on importance values. We define the energy as:

$$E_R(P') = \sum_{m_i \in M} \exp(-w_i)\ (1.0 - R(m_i, p'_i)) \qquad (1)$$

where,

$$R(m_i, p'_i) = \begin{cases} r_{f(i)} & \text{if } c^p_{f(i)} = c^m_i \\ 0 & \text{otherwise.} \end{cases}$$

The *regularization energy* aims to minimize the difference between target halftone image and the synthesized QR code. We formulate the minimization as optimizing a module-wise image difference between the assigned pattern and corresponding halftone patch while favoring the smoothness between neighboring patterns. We construct a graph $G = (V, E)$ where the nodes V correspond to the modules in M. Two nodes m_i and m_j are connected by an edge $e_{ij} \in E$, if m_j is in the 4-connected neighborhood of m_i. The regularization energy is defined as:

$$E_G(P') = \sum_{m_i \in V} D(I^m_i, I^p_{f(i)}) + \\ w_s \sum_{e_{ij} \in E} \exp(-D(I^m_i, I^m_j))\ D(I^p_{f(i)}, I^p_{f(j)}) \qquad (2)$$

where, w_s represents the weight of smoothness energy ($w_s = 0.2$ in our tests). As for the similarity distance function, $D(\cdot, \cdot)$, we use the structural similarity index measure (SSIM) introduced by Wang *et al.* [2004], and later used by Pang *et al.* [2008] to quantify the difference between gray-scale and halftone images. Since P is fixed, we further improve the performance by pre-computing the pair-wise similarity distance among all patterns, normalizing the value to [0, 1.0] and compile the results into a 512×512 matrix as shown in Figure 3.

In order to ensure that the final assignment of patterns retains and binds the original data of modules to the center pixel of pattern, we introduce a binding constraint to penalize the assignment of patterns with wrong color and is defined as:

$$E_C(P') = \sum_{m_i \in M} \beta\ \delta_{m_i, p'_i}, \qquad (3)$$

with $\beta = 100$ and δ_{m_i, p'_i} is set to 1 if $c^p_{f(i)} \neq c^m_i$ and 0 otherwise.

Solving the pattern assignment problem then amounts to minimizing the total energy,

$$E_{total}(P') = \lambda E_R(P') + E_G(P') + E_C(P') \qquad (4)$$

with λ controlling the relative importance between the reliability and regularization energies. From the useability perspective, this also allows the user to control the level of readability according to the performance of different barcode readers (see Section 6). By carefully organizing unary and binary terms in Equation 4, our pattern assignment formulation is equivalent to multi-label Markov random fields problem. We use a graph cut algorithm [Boykov et al. 2001] to minimize the total energy efficiently to obtain a desirable pattern assignment for each module.

Pattern reliability evaluation. Ideally, as long as a QR code is scanned under perfect conditions such that no module is distorted or blurred, replacing its modules by any pattern with identical color at the center pixel should not compromise its readability. However, this is rarely the case. In reality the readability of QR code easily gets affected by factors such as insufficient lighting conditions, poor camera resolution and focus capability, and the reading performance of a barcode reader. Imprecise understanding of such effects makes it difficult to analytically evaluate the readability of a visual QR code. Instead of striving to obtain a boolean value to indicate readability, we introduce the pattern reliability as a probability of retaining the module's readability by the pattern substitution.

Quantifying this uncertainty requires a sufficiently large set of QR codes embedded with designated pattern and an automatic mechanism to test and return whether the module replaced by that pattern is correctly decoded. We created a database of synthetic QR codes using an automatic procedure to generate them with different data and patterns, and apply spatial perturbation to simulate scanning the QR codes in reality. Then, we use a barcode reader to decode each synthetic QR code and statistically model the pattern reliability.

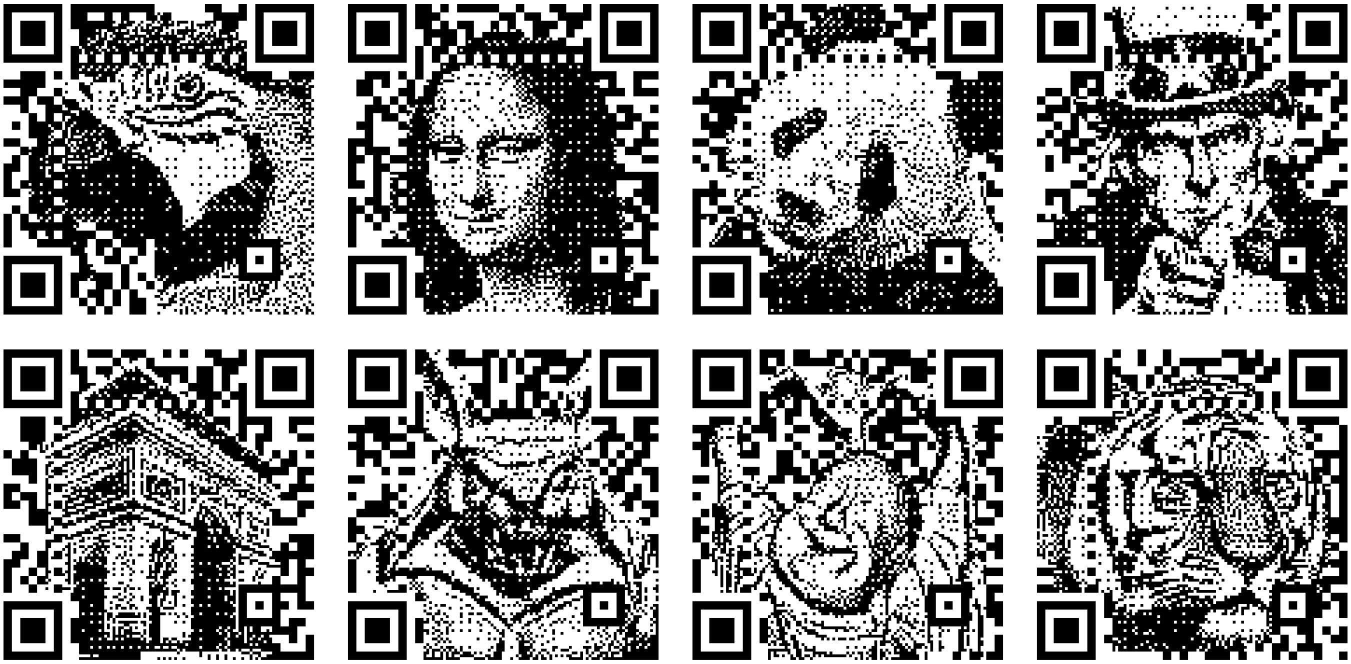


Figure 5: *Eight halftone QR codes generated by our method (the original images are shown in Figure 10).*

Synthetic QR code database: The procedure first uses the same data encoding library and random data string to generate QR codes with different appearances. Then it replaces each QR code module with a pattern randomly picked from a subset of P in which patterns have identical color to the module at center pixel (see Figure 4(a)). To evaluate how each pattern is resistant to perturbation, we apply a variety of spatial perturbation to each synthetic QR code to account for different scenarios when scanning a code in reality. Specifically, to mimic the situation of scanning the QR code from non-frontal orientations, we apply a random perturbation of yaw, pitch and translation in xy plane to QR code image. The range of rotation and translation are from $[-3, 3]$ in degree and $[-1, 1]$ in pixel, respectively. We also use a random scaling factor ranges from [1, 30] to simulate scanning the code at different distances.

Pattern reliability evaluation: We used an open source barcode reader, called ZBar [Brown 2007], to decode every synthetic QR code in database. In addition to returning a boolean value indicating the readability of the QR code, ZBar is also able to provide module-wise response that informs whether a module is decoded correctly. We calculate pattern reliability as a ratio of number of successful decodes among all samples in database and properly normalize the value. Figure 4(b) shows typical examples. Finally, we obtain a database of 0.6 million synthetic QR codes in which the variation of estimated pattern reliability is stabilized as shown in Figure 4(c). To measure stability, we calculate pattern reliability variation as norm of vector difference and use a threshold of 0.01.

5 Enhancement

Our approach is able to utilize the error correction capability of QR code to further improve the quality of results. The idea is to allow a few modules to be free from the binding constraint and permit those modules to pick the best pattern from the full set of P, thus improving the visual quality at the cost of losing data in modules which will be restored by error correction mechanism.

QR code implements the Reed-Solomon error correction algorithm that divides a QR code into several disjoint RS blocks and perform the error correction on individual block. Each RS block consists of codewords and each codeword contains eight data modules. Codeword is the basic unit in error correction, and is considered as erroneous and needed to be restored if one of its eight data modules is decoded incorrectly. QR code specification defines four error correction levels, which are L(7%), M(15%), Q(25%), and H(30%), where the number indicates the percentage of codewords that can be restored in each RS block. Therefore, it is easy to calculate the error correction capacity (ECC) of RS blocks in a QR code given a specific setting of symbol version and error correction level.

We adopt a greedy approach to iteratively remove binding constraint from codewords. First, we replace β in Equation 3 with a β_i for each module m_i and initialize its value to 100. Then for each RS block, we sort its codewords according to a priority calculated by averaging the importance weight of eight data modules. The codeword of highest priority will be set free and the β_i of its modules is set to 0. Next, we solve the pattern assignment optimization and count the number of erroneous codewords in each RS block. The process repeats until it reaches the desired number of erroneous codewords (50% of ECC in our tests).

In order to effectively distinguish semantic salient features (e.g. faces, eyes, etc.) from gradient-based features, we provide a brush

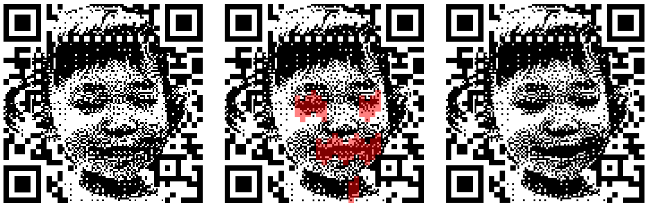


Figure 6: *(Left) Result before enhancement. (Middle) Free modules are highlighted in red. (Right) Result after enhancement.*

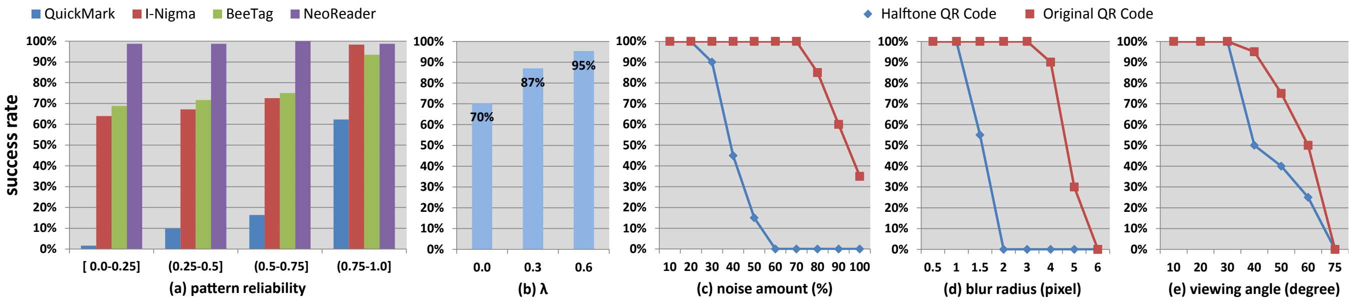


Figure 7: *(a) Success rates of barcode readers operating on different group of synthetic QR codes. (b) Success rates of scanning generated halftone QR codes at three levels of readability. ((c)-(e)) Compare success rates of scanning halftone and original QR codes under various distortions (Gaussian noise, Gaussian blur and scanning at tilted angle).*

tool for users to scribble on critical regions to guide the codewords' priority. In Figure 6, we scribble a few strokes on the eyes and mouth of girl. Since the enhancement is mainly effective around semantic features, for easy comparison, we do not use it in most of the results shown in the paper. We refer readers to the supplementary material for the comparison of results with and without the enhancement.

6 Results and Discussion

We have tested our method on a wide variety of images. Figure 5 shows eight results generated by our method and a complete gallery of halftone QR codes could be found in the supplementary material. The default QR code setting of symbol version 5 and error correction level H* is used for generating all the results. Our algorithm is computationally efficient and takes less than a second to generate a result using unoptimized codes on a PC with 3.4GHz CPU and 8 GB system memory. The performance of our algorithm is dependent on the size of QR code and the number of data modules involved in the optimization. The halftone nature makes our results gain visual quality superior to existing visual QR codes as viewed from a distance; and our representation model that shows stable image quality under the variation of QR codes also enables an effortless extension to combine QR codes with video clips to generate animated halftone QR codes. Although a simple frame-by-frame synthesis is employed without taking into account the temporal coherence between consecutive halftone images, the generated results are quite satisfactory in terms of visual quality and readability during the video playback (see supplementary video).

As the best reading performance could be obtained when QR code's modules are scanned with no distortion and blurring, determining a proper size of results that are either shown on a display device or printed on a paper becomes an importance issue. We recommend readers to enlarge the image in the digital display to obtain a clear vision of submodules before scanning the image. To determine the actual image size in the printout, there is a guideline [DENSO WAVE 2003] for measuring the actual size of module according to printer specifications. We took a conservative specification of a laser printer with 360dpi resolution and obtained a recommended submodule size of 0.35mm, indicating a recommended module size of 1.05mm. Results shown in the paper are carefully scaled to fit the estimated size as close as possible.

To validate the effectiveness of pattern reliability and controllable level of readability of generated halftone QR codes, we conducted an experiment to manually scan resultant QR codes using a smartphone equipped with 8-megapixels camera and four popular barcode readers. We also conducted an experiment to compare halftone QR code with original QR code to estimate the sensitivity of readability under various distortions. During the experiment, we showed one QR code on a monitor per trial and user who holding the smartphone was asked to scan the code. During the scan, he/she was allowed to rotate the device and move it forward and backward as we normally do, and asked to put down the device before starting the next trial to avoid a bias scan. We define a successful scan as the barcode reader returns a correct data string within 3 seconds.

Experiment I: pattern reliability. Effective pattern reliability means that patterns with lower reliability are more likely to fail the scan in reality than patterns with higher reliability. To verify this hypothesis, we uniformly divided the range of reliability into four intervals and classified patterns into four groups. For each group, we generated 60 synthetic QR codes using the patterns from the group, and each QR code was scanned by three users using four barcode readers. Figure 7(a) shows the success rates of barcode readers operating on each group. It indicates not only an expected positive correlation between pattern reliability and response of barcode readers, but also the relative performance among barcode readers. In the next experiment, we shows that by adjusting the value of parameter λ, we could effectively control the readability of generated halftone QR codes.

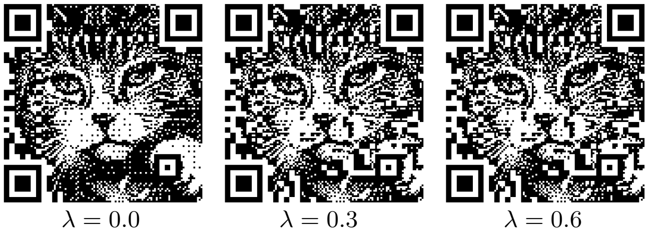

$\lambda = 0.0$ $\lambda = 0.3$ $\lambda = 0.6$

Figure 8: *Halftone QR codes at different level of readability. The readability is improved as increasing the value of parameter λ.*

Experiment II: level of readability. First, we used a default value of $\lambda = 0.0$ to generate one hundred halftone QR codes and run the manual scan procedure as in experiment I. We found that 91% of the results can be successfully read by all barcode readers, meaning that our representation model could handle large type of images with no difficulty. For the rest of 9% results which failed in one or more barcode readers, we regenerated results with $\lambda = 0.3$ and $\lambda = 0.6$, and repeated the experiment. We hypothesize that the readability of these results could be effectively improved as we increase the λ (see Figure 8). Figure 7(b) shows the average success rates against

*The parameter will change according to the length of data string.

different λ values, and a single factor ANOVA analysis with $\alpha = 0.05$ is used to verify our hypothesis (p-values < 0.01).

Experiment III: sensitivity of readability. To compare the performance of halftone QR code with original QR code, we randomly selected 5 out of 91% successful results and corresponding original QR codes, then separately applied to images Gaussian noise with various amounts from 10% to 100%, Gaussian blur with kernel radius from 0.5 to 6 pixels, and scanned the code at tilted angle from 10 to 75 degrees. Each QR code was scanned by three users using four barcode readers. As shown in Figure 7((c)-(e)), the halftone QR codes stop being functional under excessive setting of Gaussian noise (40%), Gaussian blur (2 pixels) and viewing angle (40 degrees), while the original QR codes crash at Gaussian noise (80%), Gaussian blur (6 pixels) and viewing angle (60 degrees). In fact, this tradeoff allows our algorithm to create visual QR codes.

Limitations. The visual quality of our results is restricted by the employed halftoning technique which might produce low quality images due to the low image resolution and contrast. For instance, QR codes of symbol version 1 and 2 will render to images of size 63×63 and 75×75 pixels, respectively, and both are too small to produce satisfactory halftone results as shown in Figure 9. Although we could use a sophisticated subdivision model such as 5×5 to increase the image resolution, it will exponentially increase the number of binary patterns as well and makes pattern reliability evaluation and pattern assignment optimization intractable. Besides, the readability of our results is inherently sensitive to blurred image caused by image downsampling. The center pixel of pattern which carries the original data module is easily polluted and damaged by its neighboring pixels.

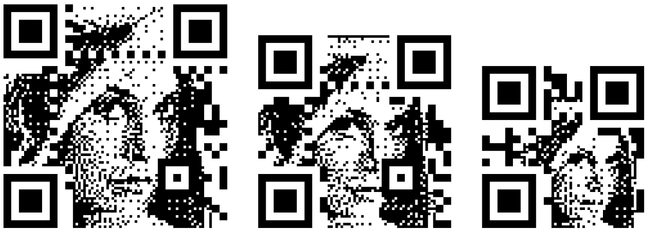


Figure 9: *Limitation: Visual quality of our result degrades as the symbol version of QR code decreases from 3(left) to 1(right).*

7 Conclusion and Future Work

Creating visually attractive QR codes has received a growing demand in diverse fields especially in marketing and advertising. The main challenge lies in how to incorporate high-level visual features (e.g. colors, letters, illustrations, logos, etc.) with ordinary QR code without compromising its readability. We presented an automatic algorithm to create a new type of visual QR code, called halftone QR code, at a controllable level of readability. There are two major components in our approach. One is a representation model that minimally binds to the original QR codes and is flexible to adapt to target halftone images; and the other is an introduction of pattern reliability which enables the control over level of readability. We proposed a systematic approach to automatically build a large database of synthetic QR codes and used it to statistically model the pattern reliability. Finally, we tested our approach using a wide variety of images and conducted an experiment of manually scanning the resultant QR codes to validate the effectiveness of our approach. In addition to generating valid halftone QR codes, we also believe that our synthetic database is a potential benchmark tool for evaluating the performance of various barcode readers.

Several interesting future works are worth exploring. First, using a color halftone image instead of a binary one would be an interesting extension to produce more attractive visual QR codes. This might be done by using a modified similarity distance metric which accounts for color difference and introducing additional term to preserve the original contrast between adjacent modules. In the application of animated halftone QR codes, we currently do not take into account the temporal coherence, resulting in the flickering artifacts. However, we realized the factor that not every frame needed to be readable during the animation, thus enabling the flexibility to incorporate temporal coherence at the cost of losing readability at some frames. Understanding a proper time interval of dropped readability without hurting user's feeling regarding to readability, and preserving temporal coherence between halftone images are challenging problems which we would like to investigate in the near future. Lastly, our approach might benefit other types of two-dimensional barcodes as well, enabling the incorporation of halftone images that produces visually appealing barcodes.

Acknowledgements. We are grateful to the anonymous reviewers for their comments and suggestions; Jr-Iang Chiou for generating the results; all the participants of the user study for their time of scanning QR codes; and Gerardo Figueroa for video narration. We are thankful to Luis Sousa for granting permission to use the panda photo. The Cat and Arc de Triomphe are image courtesy of David Corby and Benh LIEU SONG, respectively. The project was supported in part by the National Science Council of Taiwan (NSC-102-2221-E-007-055-MY3 and NSC-102-2220-E-007-023), Ministry of Economic Affairs of Taiwan (MOEA-102-EC-17-A-02-S1-202), an Adobe research gift and an UCL impact award.

References

A T Communications Co., L., 2007. LogoQnet. http://logoq.net/.

Boykov, Y., Veksler, O., and Zabih, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 11, 1222–1239.

Brown, J., 2007. Zbar bar code reader. http://zbar.sourceforge.net/.

Chan, H., 2011. HOW TO: Make Your QR Codes More Beautiful. http://mashable.com/2011/04/18/qr-code-design-tips/.

Chang, J., Alain, B., and Ostromoukhov, V. 2009. Structure-aware error diffusion. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5, 162:1–162:8.

Chu, H.-K., Hsu, W.-H., Mitra, N. J., Cohen-Or, D., Wong, T.-T., and Lee, T.-Y. 2010. Camouflage images. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 51:1–51:8.

Cox, R., 2012. QArt Codes. http://research.swtch.com/qart.

DENSO WAVE, 2003. QR code.com. http://www.qrcode.com/en/.

D'Haem, J., 2010. QR Code ®Artist. http://www.qrcartist.com/.

flickr, 2009. QR code Art. http://www.flickr.com/groups/qr-art/.

Gal, R., Sorkine, O., Popa, T., Sheffer, A., and Cohen-Or, D. 2007. 3D collage: Expressive non-realistic modeling. In *Proc. of NPAR*, ACM, 14.

HUANG, H., ZHANG, L., AND ZHANG, H.-C. 2011. Arcimboldo-like collage using internet images. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 30*, 6 (Dec.), 155:1–155:8.

ISO/IEC 18004, 2006. Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification.

KENTARO, F., 2006. QRencode. http://fukuchi.org/works/qrencode/.

KOPF, J., AND LISCHINSKI, D. 2011. Depixelizing pixel art. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4, 99:1–99:8.

KYPRIANIDIS, J. E., AND DÖLLNER, J. 2008. Image abstraction by structure adaptive filtering. In *Proc. EG UK Theory and Practice of Computer Graphics*, 51–58.

LAPORTE, A., 2012. Unitag. http://www.unitaglive.com/qrcode.

MAHARIK, R., BESSMELTSEV, M., SHEFFER, A., SHAMIR, A., AND CARR, N. 2011. Digital micrography. *ACM Trans. Graph. (Proc. SIGGRAPH) 30*, 4, 100:1–100:12.

MITRA, N. J., AND PAULY, M. 2009. Shadow art. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5, 156:1–156:7.

PANG, W.-M., QU, Y., WONG, T.-T., COHEN-OR, D., AND HENG, P.-A. 2008. Structure-aware halftoning. *ACM Trans. Graph. (Proc. SIGGRAPH) 27*, 3, 89:1–89:8.

PELED, U., 2012. Visualead. http://www.visualead.com/.

SCHEIB, V., 2010. QR Code hacks: modifying and altering for artistic fun. http://beautifulpixels.blogspot.tw/2010/08/qr-code-hacks-modifying-and-altering.html.

WANG, Z., BOVIK, A. C., SHEIKH, H. R., AND SIMONCELLI, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. on Vis. and Comp. Graphics 13*, 4, 600–612.

WINTER, M. 2011. *Scan Me: Everybody's Guide to the Magical World of QR Codes*. Westsong Publishing.

XU, X., ZHANG, L., AND WONG, T.-T. 2010. Structure-based ascii art. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 52:1–52:10.

Figure 10: *Original images of Figure 1, Figure 8, and Figure 5.*