TOWARDS ROBUST COLOR RECOVERY FOR HIGH-CAPACITY COLOR QR CODES

Zhibo Yang, Zhiyi Cheng, Chen Change Loy, Wing Cheong Lau, Chak Man Li, Guanchen Li

Department of Information Engineering, The Chinese University of Hong Kong

ABSTRACT

Color brings extra data capacity for QR codes, but it also brings tremendous challenges to the decoding because of color interference and illumination variation, especially for high-density QR codes. In this paper, we put forth a framework for high-capacity QR codes, *HiQ*, which optimizes the decoding algorithm for high-density QR codes to achieve robust and fast decoding on mobile devices, and adopts a learning-based approach for color recovery. Moreover, we propose a robust geometric transformation algorithm to correct the geometric distortion. We also provide a challenging color QR code dataset, CUHK-CQRC, which consists of 5390 high-density color QR code samples captured by different smartphones under different lighting conditions. Experimental results show that HiQ outperforms the baseline [1] by 286% in decoding success rate and 60% in bit error rate.

Index Terms— QR code, color recovery, color interference, illumination variation, high capacity

1. INTRODUCTION

Recently, there has been a dramatic increase in the usage of QR codes. However, the relatively low capacity of existing QR codes has limited their applicability, e.g. in protecting the authenticity of a QR code [2]. To increase the data capacity of QR codes, leveraging color is probably the most natural and inexpensive approach.

Due to chromatic distortion, it is challenging to decode color QR codes in a robust manner, especially for high-density ones. Possible root causes of such chromatic distortion include: 1) *cross-channel interference*: colorants in each channel tend to interfere with each other [1]; 2) *cross-module interference*: for a high-density QR code, the colorants in the neighboring data modules (spatial regions) may spill over and distort the color; 3) *illumination variation*: color varies dramatically under different lighting conditions [3]. Besides, when a QR code becomes very dense, even for a monochrome one, decoding becomes very difficult under small module sizes and the tolerance for *geometric transformation* becomes much more stringent.

There have been numerous attempts in the literature using color to increase the capacity of traditional 2D barcodes [1][4][5][6][7]. Microsoft High Capacity Color Barcode (HCCB) [5] encodes data using color triangles with a predefined color palette. The color QR code framework, HCC2D [4][8], also encodes multiple data bits in each color symbol and uses 4- and 8-color palette to provide different data capacities. H. Blasinski et al. [1] propose the per-colorant-channel color (PCCC) framework for 2D barcodes where data are encoded in the three channels of the CMY color space during printing. Shimizu et. al. [9] propose a 64-color 2D barcode and augment the RGB color space using seed colors for references. All these methods use reference colors during decoding. However, our preliminary testing shows that using reference colors can easily introduce bias during color recovery. It is because any dirt or damage on the reference symbols, as well as non-uniform lighting, can easily make color QR codes undecodable. Although the PCCC framework adopts an iterative decoding algorithm without using reference colors, it is too computationally expensive for a mobile device. Moreover, none of these methods have been evaluated using high-density color QR codes.

In this paper, a new color QR code framework, *HiQ*, is proposed. HiQ constructs a color QR code by combining multiple monochrome QR codes together in a layered manner to preserve the structure of traditional QR codes and the strength of their design. Firstly, we improve the decoding algorithm to robustly decode high-density QR codes under the HiQ framework. Secondly, we utilize a learning-based approach to model the lighting variation and color interference to tackle the chromatic distortion. Thirdly, a robust geometric transformation method is proposed to correct the geometric distortion. Finally, we use a traditional QR code decoder to extract data from each layer after colors are recovered using a pre-trained color classifier. Furthermore, we introduce a large-scale dataset of color QR codes, CUHK-CQRC, to facilitate the evaluation of our proposed algorithms.

2. HiQ: A LAYERED FRAMEWORK FOR HIGH-CAPACITY QR CODES

2.1. Data encoding

To leverage existing QR code standards and tools, HiQ partitions the data to be encoded into multiple portions and encodes them into different monochrome QR codes *independently*. We refer to each monochrome QR code as a *layer* in HiQ. Note that the monochrome QR codes of different layers

This work is supported in part by the Innovation and Technology Fund of the Hong Kong SAR Government (Project no. ITS/300/13) and the Knowledge Transfer Fund at CUHK (Project no. TBF14ENG004).

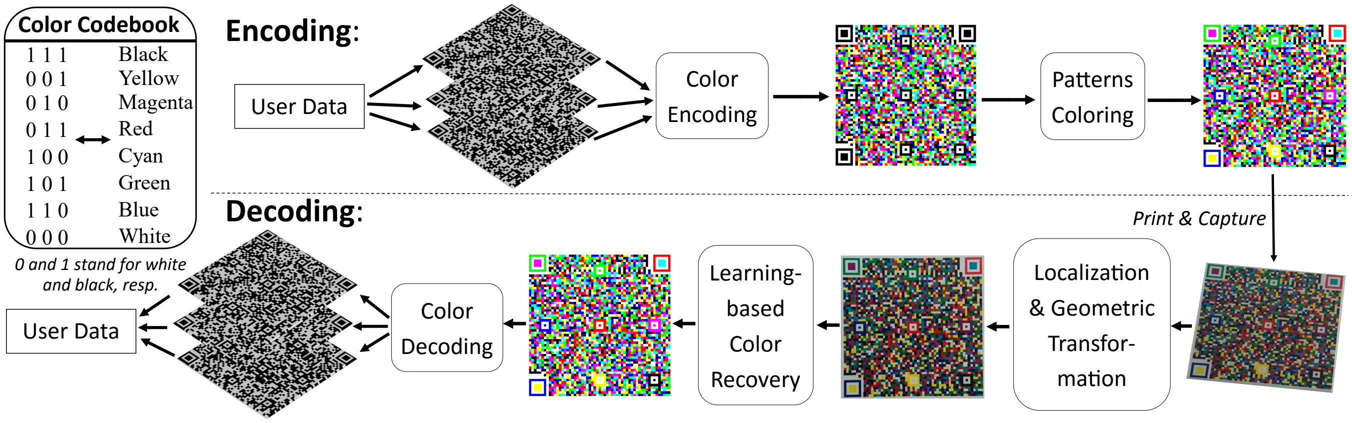


Fig. 1. An overview of the encoding and decoding of a 3-layer color QR code in HiQ.

can have different levels of error correction, but they have to be of the same dimension/ size in order to make each layer of the color QR code separable.

Fig. 1 illustrates the construction of a 3-layer color QR code. Given n monochrome QR codes, $\{M_i\}$, where $i = 1, 2, \cdots, n$, each M_i is composed of the same number of small squares (referred as modules) with two different colors (e.g., white and black) which represent one data bit, 0 and 1. We denote the jth module of M_i by m_j^i, where $m_j^i = 0$ or 1. In order to achieve layer independence and separability in the color QR code, HiQ constructs an n-layer color QR code C_n by concatenating all M_i together so that the jth module of C_n, $c_j^n = \{m_j^1, m_j^2, \cdots, m_j^n\}$. HiQ then maps each c_j^n into a particular color using a predefined color codebook $\mathbb{B}$, where $|\mathbb{B}| = 2^n$ as m_j^i is binary. Finally, special patterns are colored with specific colors to filter out false positives of pattern detection and carry meta informations of the QR code, e.g., number of layers it contains.

2.2. Data decoding

An overview of the decoding process of a 3-layer color QR code is given in Fig. 1. The decoding process is to extract user data embedded in the QR code. Typically, given an image of a monochrome/ color QR code captured by a smartphone camera, HiQ first binarizes it for detecting the special patterns which are then used to determine the boundary and correct geometric distortion of the QR code. Before decoding, HiQ will read the number of layers embedded in the special patterns/modules of the code. If a QR code contains multiple layers, i.e., a color one, HiQ will apply *color recovery* and use the predefined color codebook to map the color QR code into multiple monochrome QR codes. Finally, HiQ extracts the data from each monochrome QR code layer by layer while leveraging the built-in error correction mechanism.

Local binarization: Existing monochrome QR code decoders usually use the image luminance to binarize the QR code. However, directly applying it on color ones can be problematic because some colors (e.g., yellow) have much higher luminance than the others, which makes some patterns undetectable. To solve this problem, we use a simple but effective method to binarize the color QR codes. Let $\mathbf{I}$ denotes an image of a color QR code formatted in the *RGB* color space. We equally divide it into 8×8 blocks. In each block, a threshold is computed for each channel as follows: $T_i = (\max(\mathbf{I}_i) + \min(\mathbf{I}_i))/2$, where $i \in \{R, G, B\}$ and $\mathbf{I}_i$ is the ith channel of image $\mathbf{I}$. A pixel denoted by a triplet (P_R, P_G, P_B) is assigned black if $P_i < T_i$ for any $i \in \{R, G, B\}$, white otherwise.

Data block accumulation: During encoding, data bits are divided into dozens of data blocks of a QR code, and decoding is performed block by block. Even if only one block of data is missing, existing decoders will initiate a new round of scanning while discarding all information obtained so far. To reduce the scanning latency, we accumulate the successfully decoded data blocks in previous scans until all the data blocks are successfully decoded.

3. ROBUST COLOR RECOVERY

For a monochrome QR code, a simple local thresholding method is enough for the color recovery since there are only two colors between which the chromatic contrast is often high. However, as the number of colors in a color QR code increases (exponentially with number of layers), the color recovery becomes non-trivial, especially in the presence of the chromatic distortion as illustrated in Fig. 2.

3.1. Robust geometric transformation

Standard methods estimate the transformation matrix by detecting four special patterns in the corners of a QR code. However, in practice, the detection is inevitably inaccurate, and the cross-module interference (see Fig. 2(a) for illustration) makes the decoding more sensitive to transformation errors caused by the inexact detection, especially for a high-density QR code. Instead of developing a more complex detection algorithm which increases processing latency, we circumvent this problem by using a robust geometric transformation (RGT) algorithm which accurately samples for each module a pixel within the central region where the color interference is less severe than that along the edges of a module. Unlike standard methods, RGT leverages all special patterns, including the internal ones, and solves a *weighted* overdetermined linear system to estimate transformation matrix.

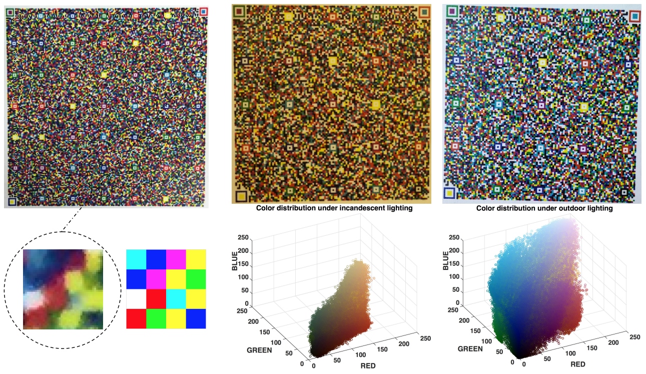
Given n tuples each consists of a pair of 2D data-points, namely, $\{< \mathbf{x_i}, \mathbf{x_i}' >, i = 1, 2, \cdots, n | n > 4\}$, where $\mathbf{x_i}$ is the position of a detected pattern, and $\mathbf{x_i}' = (x'_i, y'_i)$ is the corresponding point in the data matrix to be reconstructed. RGT estimates the transformation matrix H by minimizing the sum of the *weighted* transformation errors $\sum_{i=1}^{n} w_i \|A_i \mathrm{H}\|_2$ subject to $\|\mathrm{H}\|_2 = 1$, where w_i is the weighting factor of each pair of points and $A_i = \begin{bmatrix} \mathbf{0}^T & -\mathbf{x_i}^T & y'_i \mathbf{x_i}^T \\ \mathbf{x_i}^T & \mathbf{0}^T & -x'_i \mathbf{x_i}^T \end{bmatrix}$. The solution is just the corresponding singular vector of the smallest singular value of the matrix built from all $w_i A_i$ [10]. As we found that the estimated position of a finder pattern is often more accurate than that of a alignment pattern, we give higher weights to detected positions of finder patterns and lower weights to alignment patterns. Empirically, we set $w_i = 0.6$ if $\mathbf{x_i}$ is from a finder pattern, 0.4 otherwise.

RGT is robust to *minor* shift in the detected positions, but *not* false positives. To reduce the false positives, we take advantage of the color property by coloring each pattern with a specific color in the encoding phase (see Sec. 2.1). For each detected pattern, we filter out the false detections by checking whether the color of it is correct or not.

3.2. Learning-based color recovery

Fig. 2(b) depicts that the variation of distribution of the color values in the RGB space under different lighting conditions. Such variations give rise to the so-called color constancy [11] problem which has been (and continued to be) an active area of computer vision research. However, most existing algorithms for achieving color constancy tend to be computation intensive and thus not viable for our application of color QR code decoding using off-the-shelf smartphones. To tackle this problem, we first normalize the color values of each pixel by leveraging the structure of QR codes and adapting a fast and effective color constancy algorithm [3] to make the color less illumination-sensitive. We then use a color classifier trained off-line to model the color variations using the normalized color values as inputs.

Given a captured image of a n-layer color QR code, we first estimate the white RGB values, $\mathbf{W}$, of the captured image from the known white regions in the QR code (e.g., along the boundaries and within the special patterns). Denoted by $(\mathbf{x}, y)$ a pixel sampled by RGT, where $\mathbf{x}$ is the color feature and $y = 1, 2, \cdots, n$ is the color label. Instead of directly using the RGB values, $\mathbf{I} = \{\mathbf{I}_R, \mathbf{I}_G, \mathbf{I}_B\}$, of each pixel as the color feature for color recovery, we normalize $\mathbf{I}$ by $\mathbf{W}$ to yield $\mathbf{x} = \mathbf{I}_j / \mathbf{W}_j$, where $j \in \{R, G, B\}$. Taking $\mathbf{x}$ as the input, the classifier outputs y which is then mapped to n binary bits for decoding using the encoding color codebook, described in Sec. 2.1. We evaluate various mainstream machine learning techniques (see Sec. 4 for details) and adopt *quadratic discriminant analysis (QDA)* [12] as the color classifier. Unlike other frameworks such as [1] and [4], which train the classifier on-line for each captured image, HiQ learns the parameters of the classifier off-line using color data collected under a wide range of realistic settings. This avoids training bias and unnecessary computations on mobile devices.



(a) Cross-module color interference.

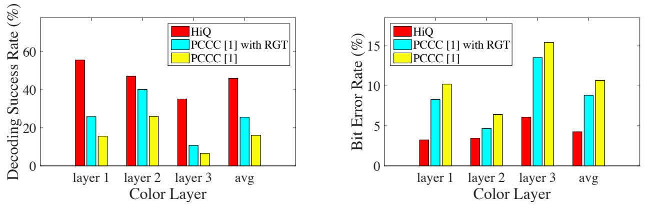
(b) Color distributions of a 3-layer color QR code under incandescent and outdoor lighting.

Fig. 2. Chromatic distortion of 3-layer color QR codes.

4. EXPERIMENTAL EVALUATION

For evaluation, we compare HiQ with the baseline method, PCCC with pilot blocks [1] over a challenging dataset, CUHK-CQRC (see the supplement for details). The performance is measured by 1) decoding success rate (DSR): $\frac{\text{\# of successful decoding}}{\text{\# of successful localization}}$, 2) bit error rate (BER): $\frac{\text{\# of recovered bits}}{\text{\# of input bits}}$ and 3) localization success rate (LSR): $\frac{\text{\# of successful localization}}{\text{\# of all test samples}}$. Since the color classifier of HiQ is trained off-line, we randomly sample 65 images of color QR codes for training which cover all different factors (e.g., different phone models, lighting conditions, etc), and use the rest for testing.

As we apply the same localization algorithms for both HiQ and PCCC, they achieve the same LSR of 68.87%. Fig. 3 shows that HiQ-C outperforms PCCC in both DSR and BER. When comparing with PCCC, HiQ increases DSR by 286%, while reducing BER by 60%. We also apply RGT on PCCC (labeled as *PCCC with RGT* in Fig. 3). RGT improves the DSR and BER of PCCC by 159% and 17.5% respectively. Across all of the 3 schemes under test, the third layer yields the worst performance, which is also consistent with the finding in [1]. This is because yellow and white under strong light, (as well as blue and black under dim light) are easily misclassified and result in a large number of classification errors (only) in the third layer. Fortunately, it is possible to



(a) Decoding success rate. HiQ surpasses PCCC in DSR by 286%.

(b) Bit error rate. HiQ reduces the BER of PCCC by 60%.

Fig. 3. Decoding results of HiQ-C, PCCC and PCCC with RGT on CUHK-CQRC.

Table 1. Color prediction performance of various methods.

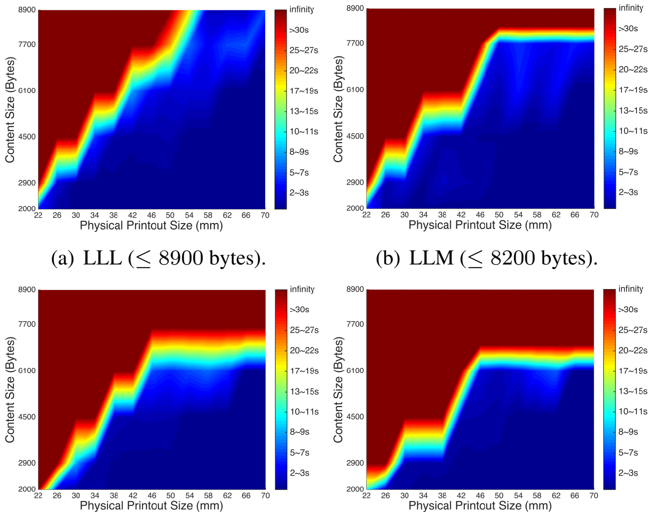
	Linear SVM	RBF SVM	QDA	RF
Accuracy	96.05%	97.67%	97.79%	96.57%

apply a higher level of Reed-Solomon error correction code during the encoding of the third layer to compensate for the higher error rate, and we investigate this possibility in Sec. 5.

We compare several classifiers for color recovery including SVM (linear and RBF kernels with one-vs-all scheme), QDA and random forest (RF) using nearly 3 million labeled pixels of 8 colors collected from color QR code images in the CUHK-CQRC. See Table 1 for the results. QDA is shown to be the most effective classifier. Furthermore, we expect training binary classifiers (e.g., SVM) for all layers independently (referred as *layered scheme*) to be a good alternative in terms of processing latency, even though it may suffer from some accuracy loss. Unlike the one-vs-all scheme which requires the training and execution of 2^n binary classifiers, the layered-scheme only needs n binary classifiers, which can reduce the real-time processing latency substantially. Our preliminary results also indicate that the accuracy of the layered scheme is comparable to one-vs-all scheme.

5. USABILITY STUDY

To evaluate HiQ in mobile applications, we implement HiQ in iOS and demonstrate the effectiveness of HiQ using a representative mobile phone, iPhone 6 Plus. We study the performance of HiQ given different content sizes (before adding error correction redundancies), printout sizes, and error correction levels. We choose four different levels of error correction denoted by, LLL, LLM, LLQ and MMM, which correspond to a maximal user-data capacity of 8900, 8200, 7600 and 7000 bytes, respectively. L, M, and Q represent Low, Medium and High level of error correction when encoding the QR code. We experiment with different levels of error correction for the third layer because it has shown to be the most error-prone layer during our previous studies. We print 3-layer color QR codes with printout sizes ranging from $22\,mm$ to $70\,mm$ and content size ranging from 2000 to 8900 bytes. We collect the scanning time of 30 successful trials for each printout under a typical lighting condition, fluorescent.



(a) LLL ($\leq$ 8900 bytes).

(b) LLM ($\leq$ 8200 bytes).

(c) LLQ ($\leq$ 7600 bytes)

(d) MMM ($\leq$ 7000 bytes)

Fig. 4. Performance of HiQ across various content sizes, printout sizes and error correction levels, measured by the 90th percentile of the scanning time of 30 successful scans. Since we cannot generate a QR code beyond its maximal capacity, we set its scanning time to infinity.

Experimental results in Fig. 4 (and those in the supplement) show that, the HiQ decoder can, in most of the cases, successfully decode the color QR code within 5 seconds. The results also present the smallest printout sizes of the color QR codes with different amount of content that can be decoded reliably and rapidly. In most cases, LLL and LLM outperform LLQ and MMM in terms of the smallest decodable printout size given the same amount of content. This suggests that higher error correction level than M on even the weak layer does not help as the use of higher level of error correction will increase the module density of the resultant QR code and makes it even more difficult to be decoded correctly.

6. CONCLUSION

This paper presented HiQ, a layered framework for high-capacity color QR codes, which supports robust and rapid decoding using off-the-shelf smartphones. HiQ enables users and developers to create generalized QR codes with flexible and broader range of choices of data capacity, error correction level and color, etc. Moreover, we have also collected a large-scale color QR code dataset, CUHK-CQRC, which will be made available to the community. Experimental results show substantial advantages of the HiQ over other baseline approaches. Our implementation of HiQ using off-the-shelf smartphones has demonstrated its usability and effectiveness in real-world mobile applications.

In this paper, we use the side-length of a square to represent the area of QR code footprint, i.e., the printout size.

7. REFERENCES

[1] Henryk Blasinski, Orhan Bulan, and Gaurav Sharma, “Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework,” *Image Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 1498–1511, 2013.

[2] Chak Man Li, Pili Hu, and Wing Cheong Lau, “Authpaper: Protecting paper-based documents and credentials using authenticated 2D barcodes,” in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 7400–7406.

[3] Arjan Gijsenij, Theo Gevers, and Joost Van De Weijer, “Improving color constancy by photometric edge weighting,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 918–929, 2012.

[4] Antonio Grillo, Alessandro Lentini, Marco Querini, and Giuseppe F Italiano, “High capacity colored two dimensional codes,” in *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on*. IEEE, 2010, pp. 709–716.

[5] Devi Parikh and Gavin Jancke, “Localization and segmentation of a 2D high capacity color barcode,” in *Applications of Computer Vision, 2008. WACV 2008. IEEE Workshop on*. IEEE, 2008, pp. 1–6.

[6] Homayoun Bagherinia and Roberto Manduchi, “A theory of color barcodes,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 806–813.

[7] Homayoun Bagherinia and Roberto Manduchi, “A novel approach for color barcode decoding using smart phones,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2556–2559.

[8] Marco Querini and Giuseppe F Italiano, “Color classifiers for 2D color barcodes,” in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*. IEEE, 2013, pp. 611–618.

[9] Takuma Shimizu, Mariko Isami, Kenji Terada, Wataru Ohyama, and Fumitaka Kimura, “Color recognition by extended color space method for 64-color 2-d barcode.,” in *MVA*. Citeseer, 2011, pp. 259–262.

[10] Richard Hartley and Andrew Zisserman, *Multiple view geometry in computer vision*, Cambridge university press, 2003.

[11] Dongliang Cheng, Brian Price, Scott Cohen, and Michael S Brown, “Effective learning-based illuminant estimation using simple features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1000–1008.

[12] Santosh Srivastava, Maya R Gupta, and Béla A Frigyik, “Bayesian quadratic discriminant analysis.,” *Journal of Machine Learning Research*, vol. 8, no. 6, pp. 1277–1305, 2007.