Mobile Color Barcode Reader

Henryk Blasinski, Sam Fok
Department of Electrical Engineering
Stanford University
Stanford, CA, USA
{h.blasinski,samfok}@stanford.edu

***Abstract*—Two dimensional machine readable symbols, i.e. barcodes, have gained popularity in recent years due to the proliferation of camera equipped mobile phones and wireless broadband internet. They are frequently used in advertising to provide customers with scannable URLs to product websites. In pursuit of increased barcode capacity, novel schemes using color have been proposed. Very few of these new schemes are extensively evaluated on mobile platforms, where limited computational resources and low image quality are limiting factors. This paper presents an Android based color QR code decoder implementation and performance evaluation on QVGA (320×240) YUV 4:2:0 encoded images. High localization $(\sim 90\%)$ and low bit error rates $(\sim 4\%)$ together with short decoding times $(< 1s)$ confirm the utility of this color QR code scheme.**

I. Introduction

Machine readable symbols, commonly known as barcodes, have been present in commercial applications for a number of years. Traditional 1D barcodes are composed of a sequence of lines with varying thickness. This variation permits the encoding of simple characters that are easy to decode via processing algorithms. Although many barcode standards exist, their capacity is typically limited to about ten digits. While this may be sufficient to uniquely represent a product identifier, as usually the case in commerce, it is insufficient to convey more complicated messages.

In response to the increased capacity requirement, novel barcode schemes have been proposed. These new designs take advantage of both image dimensions to embed data, resulting in larger capacity and information density. A typical 2D barcode is composed of a fixed pattern, independent of the encoded data, used for barcode localization in the image and a variable field used for storing specific information [1]. Usually small black and white squares or other symbols are used to represent individual bits of the embedded message, which is often secured against decoding errors with an error correction code. The most popular format for mobile applications is the QR code [2] which is very robust against localization errors, but has somewhat lower information density compared to other schemes. Other popular designs include the Aztec code [3], frequently applied in electronic ticketing and the DataMatrix [4] often used in the industrial parts labeling. According to current 2D barcode encoding standards all of these codes can carry as many as 2,000 characters, but, by doing so, their physical size increases considerably. In practice, encoded strings rarely exceed 30 characters. They are usually placed in printed advertisements and represent a URL of a product or manufacturer's website. Such a code provides a simple interface between the physical and digital worlds [5].

In order to increase the capacity even further, using color as the third information encoding dimension was proposed [6]. As pointed out in [7] the biggest obstacle in using color information is the dependence of the estimated RGB color components on the illumination. Currently, several color barcode encoding schemes have been proposed [8]–[12], however only the Microsoft High Capacity Color Barcode [13] is available as a commercial product. Recently a novel color QR code scheme was described [14]. In brief, the design consists of three independent codes printed in overlay using subtractive primaries: cyan, magenta and yellow. The goal of the decoding is to estimate the presence or absence of a given print colorant at a given spatial location from the captured RGB image data. Once binary barcodes embedded in cyan, magenta and yellow are estimated, a regular, monochrome 2D barcode reader can extract the information.

Quantitative performance of the above scheme, presented in [15], is conducted using high resolution, 24 bits per pixel, images. In mobile applications however, computational complexity is the most limiting factor. Consequently, in order to achieve reasonable processing times, it is necessary to limit captured frame sizes to 320×240 (QVGA) or 640×480 (VGA). Additionally, the camera preview stream usually provides data in the YUV 4:2:0 format, where the color information is subsampled by a factor of two, further reducing the quality of available images. In this paper we present an Android based, color QR decoding application which successfully operates on low resolution, low chroma sampling rate images.

The paper is organized as follows: section II describes the design and the decoding algorithm of the color QR code. Android implementation details are presented in section III. They are followed by experimental results in section IV and conclusions in section V.

II. Color Barcode Design

The color QR code design closely follows the scheme proposed by [14]. The code is composed of three independent QR codes printed on top of each other using subtractive color primaries: cyan, magenta and yellow (Fig. 1). During acquisition, barcode colors are expressed as linear combinations of additive primaries: red, green and blue. Theoretically, red is

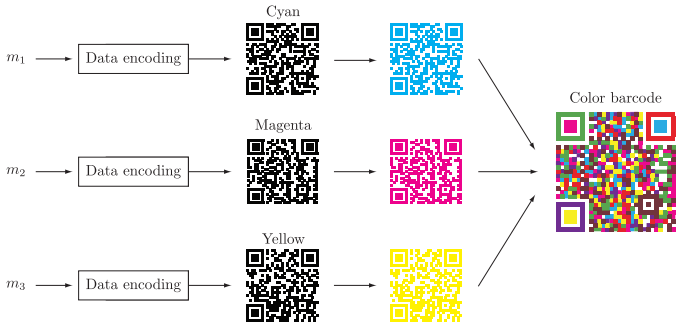

Fig. 1. Color QR code as an overlay of monochrome codes printed in overlay using subtractive color primaries. Each message m_i is completely independent.

a combination of magenta and yellow and therefore can be interpreted as 'negative' cyan, similar analogies can be made for green and blue. In this idealistic scenario RGB channels could be used to perfectly estimate the presence or absence of a given print colorant. However in practice, as demonstrated by [16], there is a significant amount of inter-channel cross-talk making this model an oversimplification.

The inter-channel interference model was described in detail by [16]. Assuming that print colorants do not cause light scattering and that sensor spectral sensitivities are narrow-band, i.e. can be approximated with Dirac delta functions placed at specific locations in the spectrum. The colorant density $d_k(x,y)$ at the spatial location (x,y) in the kth $(k \in \{R,G,B\})$ sensor channel is related to the print colorant $I_i(x,y)$ at the same location (x,y), $(i \in \{C,M,Y\})$ by:

$$d_k(x,y) = \sum_{i \in \{C,M,Y\}} d_k^i I_i(x,y) \tag{1}$$

The optical density $d_k(x,y)$ can be estimated using:

$$d_k(x,y) = -\log_{10}\left(\frac{I_k^s(x,y)}{I_k^s(W)}\right) \tag{2}$$

where: $I_k^s(x,y)$ is the captured image value observed in the kth channel and $I_k^s(W)$ is the captured image value corresponding to the white point in the kth image channel. If all nine linear weights d_k^i $(i \in \{C,M,Y\}, k \in \{R,G,B\})$ from (1) are known, it is possible to compute the print colorant channel estimate: $I_i(x,y)$.

The proposed color QR code architecture takes advantage of localization marks that remain constant, irrespective of the embedded data. These patches are printed using specific color mixtures in known locations, these mixtures are: cyan, magenta, yellow, cyan+magenta (blue), cyan+yellow (green), magenta + yellow (red). After image acquisition and barcode localization these color templates are used to represent $I_k^s(x,y)$. The white point is estimated using data from within the localization marks. Optical densities are computed for every of the six palette colors, for all k color channels $(k \in \{R,G,B\})$, they can be arranged in a 18×1 vector, denoted $\mathbf{d^s}$. For these reference patches, the binary variable I_i describing the presence or absence of the ith colorant channel is also known and can be arranged into a 18×9 matrix $\mathbf{I}$. The unknown mixing weights d_k^i can be rearranged to form a vector $\mathbf{x}$. As these variables represent mixing weights of print colorants, they cannot be negative. Finally, the relationship (1) can be expressed in a matrix form.

$$\begin{aligned} \mathbf{Ix} &= \mathbf{d^s} \\ s.t. \quad & \mathbf{x} \geq 0 \end{aligned} \tag{3}$$

In order to find the coefficient vector x the above problem can be re-forumlated as a least squares, convex optimization task.

III. Algorithm

The barcode decoding algorithm was implemented in Java and designed for the Android mobile operating system. Fig. 3 presents the block diagram of the major decoding steps. Chroma subsampled YUV 4:2:0 encoded data from the camera is converted and interpolated to form a full resolution RGB image. The RGB data is subsequently white balanced using the gray world algorithm. Corrected image is adaptively thresholded to account for local illumination variation producing a binarized image for localization and geometric distortion removal. The convex problem of the estimating the d_k^i weights is solved using conjugate gradient descent with logarithmic barrier method. These weights are subsequently used to estimate colorant presence in CMY channels. Each CMY channel is then processed independently and treated as a separate barcode which is decoded using monochrome QR code decoding software as discussed in section II. This implementation used the open source, Java based, ZXing[1] decoder. All matrix operations were performed with the EJML[2] library.

IV. Results

The decoding software was implemented and tested using two smart-phones: Motorola Droid (600 MHz CPU, 256 MB DDR SDRAM, 5.0 MP camera) and a Motorola DroidX (1GHz CPU, 512 MB DDR SDRAM, 8.0 MP camera). Target color QR codes were printed with four error correction levels (L, M, Q, H) and two feature dimensions: 0.75 and 1.5 mm, using an HP Color LaserJet CP3525 laser printer. As discussed in [15] printer variability has little influence on the final decoding performance. Each code was acquired five times with each phone from a distance of about 15 cm in daylight and incandescent illumination. A single acquisition consisted in analyzing 30 QVGA (320×240) preview frames, giving a total of 4,800 analyzed frames. Additionally, in order to analyze the execution time, the decoding algorithm was run 20 times on a VGA resolution image (640×480) on each phone.

To quantify performance, four performance metrics were defined as follows:

1) Synchronization Success Rate (SSR): Fraction of frames in which a QR code was correctly localized.
2) Bit Error Rate (BER): Fraction of incorrectly read bits in synchronized frames.

[1] http://code.google.com/p/zxing/
[2] http://code.google.com/p/efficient-java-matrix-library/

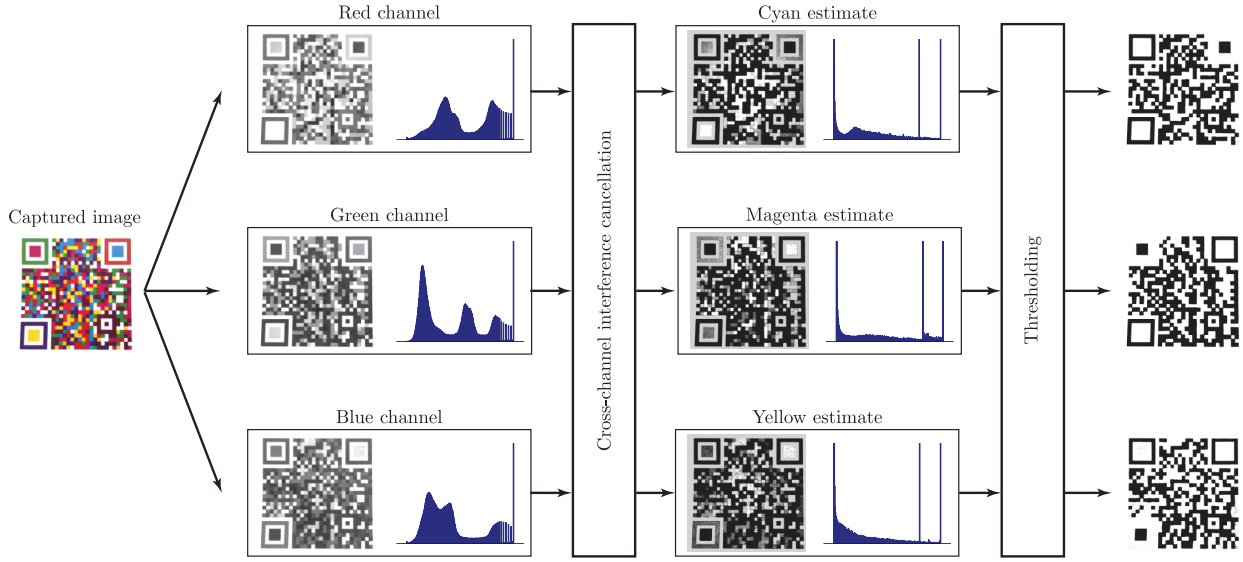

Fig. 2. Color QR decoding process. Cyan, magenta and yellow layers are estimated via cross-channel noise cancellation which increases the separation between foreground and background pixel values (see histograms) facilitating thresholding operation.

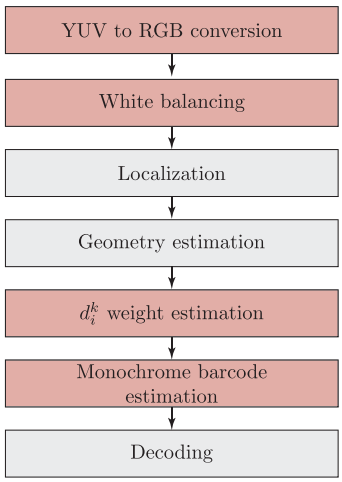

Fig. 3. Block diagram of color QR code decoding. Gray blocks indicate steps of conventional QR code decoding, and red blocks indicate additional steps required for color QR code decoding.

3) Decoding Success Rate (DSR): Fraction of synchronized frames where the embedded data were correctly decoded, using error correction code.
4) Trial Success Rate (TSR): Fraction of 30 frame long trial sequences where data in all three channels were decoded at least once.

A. Decoding accuracy

Performance metrics for 0.75 mm and 1.5 mm barcodes are summarized in tables II and I respectively. Synchronization rates ranged from a high of 87% of the frames (daylight with 1.5 mm sized features and error correction levels L and M) to a low of 22% of the frames (incandescent light with 0.75 mm sized features and error correction level Q and H). Not surprisingly, synchronization rates fell with the smaller feature size. Our experiments also demonstrate a lower decoding performance in incandescent light, which produces a slightly yellow tint interfering with the yellow reference color patch. However, synchronization rates also fell with increasing error correction levels. We can attribute the drop in synchronization rates with error correction level to the increased number of features with higher error correction levels. With a higher number of features and a fixed preview resolution, the number of camera pixels per features drops with higher error correction levels leading noisier feature estimation.

Bit error rates remained below 10% across all conditions, and once synchronized, QR code decoding rates performed above 80% across all conditions except for the yellow channel in incandescent, 0.75 mm feature, L error correction level condition. Here, yellow is very similar to background and the lowest error correction level may be insufficient to compensate.

The trial success rate provides a good indicator of user experience. If the user maintains the QR code within the camera preview for 30 frames (15 seconds at 2 Hz frame processing rate), our framework need not decode every code at once on every frame. Once a color channel has been decoded, we simply store the result and continue processing subsequent frames for the remaining codes. For a 30 frame sequence, we decoded all three channels at least once 100% of the time in in all conditions except incandescent with error correction levels Q and H. We attribute the drop in trial success rate to the same reason QR code synchronization rates fall with increasing error correction level as higher error correction levels result in fewer camera pixels per feature.

B. Execution time

We also analyzed the algorithm execution time on VGA and QVGA frames, average processing times per frame are

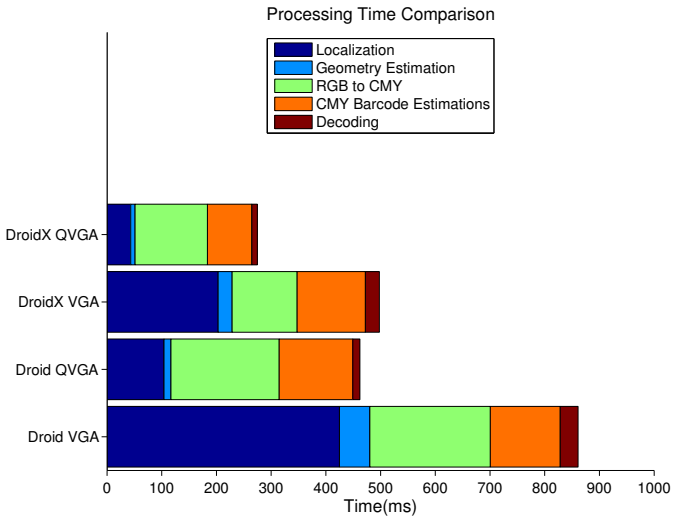

Fig. 4. Breakdown of processing times for Motorola DroidX and Motorola Droid for QVGA at QVGA and VGA resolutions. Barcode localization is most sensitive to resolution scaling while other processing steps maintain relatively fixed times.

summarized in Fig. 4. A twofold increase in computation time was observed during switching from QVGA to VGA resolutions. On closer inspection of the timing breakdown, as presented in Fig. 3, the majority of time increase was attributed to barcode localization. The remainder of operations represent a fixed overhead and are not the prohibitive factor in scaling the algorithm to larger image resolutions.

V. Conclusions

Our results demonstrated the scalability and reliability of the color barcode framework proposed in [14]. We successfully implemented color QR code decoding on a mobile device with limited computational power, YUV 4:2:0 preview frame encoding, and small preview resolution. In addition, our implementation demonstrated robustness against color variation and local lighting variation. Further design improvements can be realized by optimizing the localization step of our algorithm and improving color balancing. As noted in Fig. 4, the QR code localization is the computationally limiting factor in upscaling frame resolution, and an implementation of color balancing schemes based on common, known illumination settings such as in incandescent and fluorescent lighting should improve robustness in those scenarios.

VI. Appendix

Henryk Blasinski: Developed the interference cancellation code and the Android App, performed the experiments, prepared the poster, co-authored the final report.

Sam Fok: Implemented locally adaptive thresholding and white balancing, conducted experimental measurements, analyzed the data, co-authored the final report.

References

[1] H. Kato, K. Tan, and D. Chai, *Barcodes for Mobile Devices*. New York, NY, USA: Cambridge University Press, 2010.

[2] M. Hara, M. Watabe, T. Nojiri, T. Nagaya, and Y. Uchiyama, "Optically readable two-dimensional code and method and aparatus using the same," United States Patent 5,726,435, 10 Mar. 1995.

[3] A. Longacre and R. Hussey, "Two dimensional data encoding structure and symbology for use with optical readers," United States Patent 5,591,956, 7 Jan. 1997.

[4] C. Sant'Anselmo, R. Sant'Anselmo, and D. C. Hopper, "Identification symbol system and method with orientation mechanism," United States Patent 5,612,524, Mar. 1997.

[5] M. Ebling and R. Caceres, "Bar codes everywhere you look," *IEEE Pervasive Comp.*, vol. 9, no. 2, pp. 4–5, Apr.–Jun. 2010.

[6] E. Schubert and A. Schroeder, "Vision-based reading system for color-coded bar codes," *Proc. SPIE: Machine Vision Applications in Industrial Inspection IV*, vol. 2665, no. 1, pp. 59–68, 1996.

[7] K. Tan, D. Chai, H. Kato, and S. Ong, "A color 2D-barcode for mobile applications: Design tips," *IEEE Pervasive Comp.*, vol. PP, no. 99, p. 1, 2011.

[8] W. Claycomb and D. Shin, "Using a two dimensional colorized barcode solution for authentication in pervasive computing," in *Proc. ACS/IEEE Intl. Conf. on Pervasive Services*, Jun. 2006, pp. 173–180.

[9] H. Kato, K. T. Tan, and D. Chai, "Novel colour selection scheme for 2D barcode," in *Intl. Symp. on Intelligent Sig. Proc. and Comm. Systems (ISPACS)*, Jan. 2009, pp. 529–532.

[10] S. K. Ong, D. Chai, and K. T. Tan, "Performance of mobile camera on colour 2D barcode," in *Intl. Symp. on Intelligent Sig. Proc. and Comm. Systems (ISPACS)*, Jan. 2009, pp. 497–500.

[11] ——, "The use of border in colour 2D barcode," in *Intl. Symp. on Parallel and Distributed Processing with Applications (ISPA)*, Dec. 2008, pp. 999–1005.

[12] K. Nurwono and R. Kosala, "Color quick response code for mobile content distribution," in *Proc. of the 7th Intl. Conf. on Advances in Mobile Computing and Multimedia (MoMM)*, Dec. 2009, pp. 267–271.

[13] D. Parikh and G. Jancke, "Localization and segmentation of a 2D high capacity color barcode," in *IEEE Workshop on Applications of Computer Vision (WACV)*, Copper Mountain, CO, USA, Jan. 2008, pp. 1–6.

[14] O. Bulan, H. Blasinski, and G. Sharma, "Color QR codes: Increased capacity via per-channel data encoding and interference cancellation," in *Proc. IS&T/SID Nineteenth Color and Imaging Conference: Color Science and Engineering Systems, Technologies, and Applications*, San Jose, CA, 7-11 Nov. 2011, pp. 156–159.

[15] H. Blasinski, O. Bulan, and G. Sharma, "Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework," *IEEE Trans. Image Proc.*, 2012, (under review).

[16] B. Oztan and G. Sharma, "Per-separation clustered-dot color halftone watermarks: Separation estimation based on spatial frequency content," *J. Electronic Imaging*, vol. 19, no. 4, pp. 043 007–1–22, Oct.-Dec. 2010.

TABLE I

COLOR QR CODE PERFORMANCE UNDER DIFFERENT LIGHTING CONDITIONS, 1.5 MM FEATURE SIZE.

		EC level: L			EC level: M			EC level: Q			EC level: H		
		C	M	Y	C	M	Y	C	M	Y	C	M	Y
Daylight	SSR	0.87			0.87			0.60			0.74		
	BER	0.009	0.003	0.009	0.008	0.003	0.008	0.018	0.014	0.014	0.009	0.007	0.020
	DSR	0.81	0.98	0.88	0.99	0.99	0.98	0.94	0.96	0.96	0.97	0.94	0.88
	TSR	1.0			1.0			1.0			1.0		
Incandescent	SSR	0.68			0.72			0.22			0.35		
	BER	0.014	0.010	0.026	0.010	0.003	0.013	0.046	0.062	0.040	0.051	0.050	0.059
	DSR	0.84	0.88	0.51	0.98	0.98	0.91	0.85	0.81	0.83	0.87	0.87	0.84
	TSR	1.0			1.0			0.9			0.7		

TABLE II

COLOR QR CODE PERFORMANCE UNDER DIFFERENT LIGHTING CONDITIONS, 0.75 MM FEATURE SIZE.

		EC level: L			EC level: M			EC level: Q			EC level: H		
		C	M	Y	C	M	Y	C	M	Y	C	M	Y
Daylight	SSR	0.52			0.64			0.29			0.41		
	BER	0.023	0.021	0.039	0.032	0.029	0.054	0.042	0.044	0.088	0.030	0.026	0.068
	DSR	0.67	0.66	0.16	0.77	0.80	0.13	0.66	0.71	0.08	0.82	0.85	0.36
	TSR	0.9			0.6			0.5			0.6		
Incandescent	SSR	0.69			0.76			0.63			0.49		
	BER	0.018	0.040	0.054	0.030	0.034	0.043	0.065	0.068	0.072	0.049	0.097	0.011
	DSR	0.66	0.29	0.09	0.68	0.58	0.33	0.50	0.29	0.25	0.52	0.44	0.48
	TSR	0.5			0.9			0.8			0.6		