UC Santa Cruz

UC Santa Cruz Previously Published Works

Title

A Novel Approach for Color Barcode Decoding Using Smartphones

Permalink

<https://escholarship.org/uc/item/01k5b8dh>

Authors

Bagherinia, Homayoun
Manduchi, Roberto

Publication Date

2014-10-01

Peer reviewed

A NOVEL APPROACH FOR COLOR BARCODE DECODING USING SMART PHONES

Homayoun Bagherinia, Roberto Manduchi

University of California, Santa Cruz

ABSTRACT

The use of colors increases the information storage capacities in barcodes. Increasing the number of colors to encode information makes the decoding a challenging task due to the dependency of the surface color on the illuminant spectrum, viewing parameters, printing device and material, color fading in addition to other nuisance parameters. A popular solution is the use of a color palette of reference colors printed with barcode. The decoding becomes more challenging if a mobile phone as a decoding device is used due to the capture of images from different distances as well as angles. In addition, the barcode images are often blurry because of incorrect focus or camera shake. We present an iterative decoding algorithm that decodes the colors of all barcode patches across the barcode by minimizing the overall observation error. Our method is able to decode colors in presence of blur using a small number of colors yet ensuring high information density.

Index Terms— Color barcode decoding, color mixing, blur problem

1. INTRODUCTION

Color barcodes accessible by mobile devices become popular as an inexpensive computing tool for information encoding. The information density of conventional black and white barcode technologies is limited because only one bit per symbol is possible. Color barcode systems such as HCCB (Microsoft’s High Capacity Color Barcode) [1] increase the information density by using more colors (e.g. four or eight colors). Increasing the number of colors to encode information makes barcode decoding a challenging task. This is due to the fact that the observed color patch depends on several factors such as unknown illuminant, viewing parameters, printing device and material, and color fading in addition to other nuisance parameters. A mainstream approach to ensure robust decoding is to use a color palette of barcode colors printed with the barcode. The color palette contains all reference colors used to generate the barcode. Since the variation of a patch color and of the corresponding reference color can be assumed to be consistent due to changes such as device used, printer, and media, the patches of the barcode can be decoded robustly. For example, the HCCB method [1] clusters the colors, and assigns each cluster to one of the reference colors in the palette using the minimum distance. One challenge facing such color clustering methods using a large number of colors is that the color distribution of one class can overlap significantly with color distributions of other classes. The clustering techniques may not work well in case of blur-induced color mixing from neighboring color patches.

Decoding a color barcode using mobile devices is even more challenging. In mobile device applications, the printed color barcode is captured from different distances and angles. The barcode images are more often out of focus since the device may focus in different

part of the image when the image is captured from relatively long distance. Motion blur caused by camera shake while capturing an image contributes to decoding error. In addition, barcode localization in the captured image can cause inaccurate patch extraction in the barcode. Any or a combination of these factors could lead to incorrect decoding.

In this contribution, a new technique for decoding the color information in presence of blur is presented where the blur could be caused by incorrect focus, uniform linear motion, long-distance image capture, inaccurate barcode boundary identification, and perspective projection. Decoding error rate using mainstream methods in presence of blur maybe high since each measured patch color is a linear combination of the colors of the patch itself and its neighboring patches. Rather than decoding or classifying each individual patches independently, our iterative algorithm decodes the colors of all patches across the barcode image by minimizing the overall observation error. We show that our method enables to encode 20% to 40% more information in presence of blur using only three colors for the same barcode area than the technologies using minimum distance method.

2. RELATED WORK

The HCCB decoding method [1] classifies N clusters in color space using mean shift clustering, and assigns each cluster center to one of the reference colors in a color palette, printed with the barcode.

Wang and Manduchi [2] developed a series of algorithms to decode printed color patches by using small number of known reference colors, observed under the same unknown illuminant as the colors to be decoded. Known reference color patches enable an estimate of illuminant as a parametric color transformation between a canonical illuminant and the unknown illuminant. This transformation is used to render the unknown color patch under the canonical illuminant, which can be decoded using the training data.

Bagherinia and Manduchi [3] showed a study of the variation of small groups of printed colors and modeled their evolution due to changing illuminant using a low-dimensional linear subspace by using large number of colors ($N = 24$). Their method classifies the colors without using reference colors. They also expanded their approach [4] by considering a small number of reference colors to maximize content space of the color barcode.

Blasinski and et al [5] use a model-based interference cancellation procedure to recover the encoded data from each color channel. They propose a color interference cancellation algorithm that estimates the cyan (C), magenta (M), and yellow (Y) print colorant Channels from the Red (R), Green (G), and Blue (B) channels of the captured barcode. The encoded data is then extracted from each estimated colorant channel.

Grillo and et al [6] introduced the High Capacity Colored Two Dimensional (HCC2D) QR-based code (Quick Response), a new 2D code which increases the code data density. HCC2D in-

Fig. 1. Left-top: an example for a QST. Left-bottom: the same QST extracted from the image of a barcode. Right: examples of barcode images after perspective projection correction taken at multiple distances and viewing angle.

creases the storage space by using 4, 8, or 16 reference colors in the palette.

Querini and Italiano [7] investigated the decoding error rate on HCC2D color barcodes by using different classifiers such as Minimum Distance, Decision Trees, K-means, Naive Bayes, and Support Vector Machines (SVM) classifiers. They showed that K-means algorithm is the most effective classifier in their experiments.

None of these previous decoding approaches have addressed color barcode decoding in presence of other source of error such as blur-induced color mixing from neighboring color patches. We have proposed a method that is able to decode a barcode in presence of blur using a small number of colors ensuring high information density.

3. COLOR BARCODE AND BLUR PROBLEM

In our color barcode decoding approach, the patches in a color barcode are created from N reference colors. A color barcode can be defined as a spatial pattern of n color patches for information encoding and a color palette of rN reference color patches, resulting in a $K = n + rN$ element barcode.

Figure 1 shows examples of a color barcode with triangular color patches imaged from multiple distances and viewing angles. This barcode is made of $m \times k$ quarter square triangles (QST). A QST is a square made up of four triangles (left, top, right, bottom) whose colors are selected from N reference colors as shown in Figure 1 (left). In our decoding method, we assume that N reference colors are displayed with the barcode as $N \times QST$. Each triangular color patch is surrounded by three other triangular patches with the same or different colors.

When a barcode is observed, a robust decoding can be achieved if the distance between N clusters resulted from patch colors of the barcode is significantly large. Then a patch can be decoded by comparing its measured color to one of the measured reference colors in the palette using minimum distance. In contrast, when a color barcode is observed in presence of blur, then the measured color does not represent the color of the patch. In this case, the measured color is a mixed color resulted from the linear combination of the patch color itself and its neighboring patches based on the assumption that the device optical system is linear (Figure 1 left-bottom). A clustering technique and minimum distance classifier may not perform well in this situation. In general, blur can be caused by several factors such as incorrect focus, motion, long-distance image capture,

inaccurate barcode boundary identification, and perspective projection or a combination of these factors.

The advantage of using triangular color patches is that each patch has only three neighboring patches and a linear mixing of only four colors is created as a result of the blur. Selection of patch shapes with larger number of edges may lead to higher decoding error rate due to mixing of more colors from neighboring patches.

4. DECODING METHOD

Let the 3-dimensional vector $\bar{c}_i = [\bar{c}_i^r, \bar{c}_i^g, \bar{c}_i^b]^T$ be the measured color of the triangular patch i with (R,G,B) color vector resulted by linear color mixing. When a color barcode is observed from a relatively large distance, our observation model is such that the measured color at each triangular patch ($\bar{c}_i$) is a linear combination of the patch color itself (c_i) and the colors of its three neighboring patches ($c_{i,1}, c_{i,2}, c_{i,3}$) with unknown coefficients as

$$\bar{c}_i = w c_i + w_1 c_{i,1} + w_2 c_{i,2} + w_3 c_{i,3} \quad (1)$$

$$w = 1 - (w_1 + w_2 + w_3) \quad (2)$$

where $\mathbf{v} = [w, w_1, w_2, w_3]^T$ are the linear coefficients. The index 1 for a triangular patch is associated with the horizontal or vertical edge depending on the patch position in a QST. The indices 2 and 3 are associated with two other edges clock-wise starting from 1 followed by 2 and 3. The linear coefficients are calculated differently based on the position of a triangular patch in a QST. Let $\mathbf{w} = [w_l, w_t, w_r, w_b]^T$ be the amount of blur induced by left, top, right and bottom sides of the barcode. In case of square patches we could use $\mathbf{w}$ as the linear coefficients associated with four neighboring patches. In case of triangular patches the linear coefficient associated with each edge is calculated as the average of two elements of $\mathbf{w}$ depending on the direction of edge normal components as described in Equation 3 for left, top, right, and bottom triangular patches in a QST.

	w_1	w_2	w_3	
left	w_l	$(w_r + w_t)/2$	$(w_r + w_b)/2$	
top	w_t	$(w_r + w_b)/2$	$(w_l + w_b)/2$	
right	w_r	$(w_l + w_b)/2$	$(w_l + w_t)/2$	
bottom	w_b	$(w_l + w_t)/2$	$(w_r + w_t)/2$	(3)

Our assumption is that the amount of blur induced in different directions by $\mathbf{w}$ are less than certain value and constrained to be uniform

Fig. 2. Left: the empirical probability of decoding error vs. the number of pixels per patch. Right: the probability of decoding error of 81 patches ($B = 128$ bits) vs. the approximate barcode-camera distance. Red: images taken in fronto-parallel view. Black: all images in the data set used. ‘x’: our method. ‘•’: minimum distance method.

across the barcode image if the blur is assumed to be uniform as well.

In our decoding approach, rather than decoding the individual patches independently, we decode the colors of the patches that best justifies observed colors of all patches. We know that each patch affects the colors of all its neighboring patches. For this reason, we cannot determine the color of an individual patch without considering all patches in the barcode. An iterative technique that decodes the colors of all patches across the barcode image by minimizing the overall observation error (decomposed in a sum of individual observation errors) can solve this decoding problem. The following iterative algorithm summarizes the decoding a color barcode:

1. Determine the colors $\mathbf{e}_k$ (for $k = 1 \dots N$) of the color palette and measure the color $\bar{\mathbf{c}}_i$ of individual patches.
2. Initialize the color $\mathbf{c}_i$ of the patch i by assigning a color from the palette with index k that minimizes the distance to $\bar{\mathbf{c}}_i$

$$\min_k \|\bar{\mathbf{c}}_i - \mathbf{e}_k\| \quad (4)$$

3. Select a set of values for $\mathbf{w}$ and compute the linear coefficients $\mathbf{v}$ using Eq. 3 and 2.
4. Compute the new color of each patch using Eq. 1

$$\hat{\mathbf{c}}_i = (\mathbf{c}_i - (w_1 \mathbf{c}_{i,1} + w_2 \mathbf{c}_{i,2} + w_3 \mathbf{c}_{i,3})) / w \quad (5)$$

5. Update the color $\mathbf{c}_i$ of the patch i by assigning a color from the palette with index k that minimizes the distance to $\hat{\mathbf{c}}_i$

$$\min_k \|\hat{\mathbf{c}}_i - \mathbf{e}_k\| \quad (6)$$

6. Compute the residual error $h = \hat{\mathbf{c}}_i - \mathbf{c}_i$.
7. Repeat steps 4 to 6 until no changes in the residual error h observed.
8. Repeat steps 3 to 7 for a new set of values for $\mathbf{w}$.
9. Select $\mathbf{w}$ and the colors assigned to the patches such that the sum of the squared residuals $\sum_j h_j^2$ is minimized.

5. EXPERIMENTS

For our experiments, we created a color barcode of $3.2 \times 3.2 \text{ cm}^2$ with $N = 3$. This barcode has 324 patches. The last three QST represent the color palette of the barcode. We printed this test barcode on a paper by a regular printer. Our camera is a 8 Megapixel smart phone camera (DROID RAZR M) with JPEG image format. The user has no control over the automated non-linear (e.g. gamma) and white-point correction applied by the camera. We took 715 images from our barcode under realistic illumination conditions, viewing angles (ranging from $\pm 30^\circ$ with respect to the normal to the barcode surface), and multiple distances ranging approximately between 55 to 140 cm. The quality of barcode images can vary from one image to another due to motion, incorrect focus, and camera-barcode distance, JPEG format, and automated corrections by the camera (Figure 1).

The number of images taken in fronto-parallel view with correct focus is 351 and with incorrect focus is 105. The number of images under perspective projection with correct focus is 224 and with incorrect focus is 35. We extracted the barcodes from the images by selecting a rectangular region of interest manually containing the barcode. Canny edge detection is used to identify the edges of the barcode which can be used to undo the perspective projection of the barcode image to a square shaped barcode. We extracted each patch based on the knowledge about the pixel dimension of the barcode image and the number of the patches arrangement in the barcode.

The choice of N colors to build the barcode is relatively trivial. We have chosen the colors of the palette as Red, Green, and Blue. These colors have the largest distance from each other in the RGB color space. Red, Green, and Blue are used as the primary for additive combination of colors, as in superimposing of projected lights or in CRT displays.

We evaluated our algorithm and the minimum distance method using all 715 barcode images. We used the average patch pixels in our method and the average of 3×3 central patch pixels in the minimum distance method to compare with the reference colors in the palette. Using all pixels in the patch would lead to much higher decoding error in minimum distance method due to mix of colors on the patch borders. In contrast using 3×3 central patch pixels in our method would lead to higher decoding error since the average color

of central patch pixels does not represent the mix color resulted from the color of the patch itself and the colors of its three neighboring patches due to the uniform blur in the entire barcode image.

To determine the search range in $\mathbf{w}$, we run the algorithm on 100 images randomly selected from our data set. We selected a range of $[0, 0.5]$ for each of elements of $\mathbf{w}$ with a step size of 0.0357 to reduce the computation time due to a 4 degrees of freedom. Then, we select the range of $\mathbf{w}$ to $[0, 0.15]$ with the same step size by determining the min and max of all elements of $\mathbf{w}$ found for these 100 images. The average number of iterations at step 7 of the algorithm is 3. The execution time to compute the $\mathbf{w}$ for a barcode is on average around 7 seconds using Matlab on Intel i5-2520M CPU @ 2.50GHz.

To evaluate and compare the performance of our algorithm with the minimum distance method, we define the empirical probability of decoding error $P(p)$ for each patch size p (number of pixels per patch) as described in [3]. $P(p)$ is defined as the number of color patches, which were incorrectly decoded, divided by the number of the barcode images (with the same patch size), and by the number of color patches in the barcode (312). Figure 2 (right) shows the empirical probability of decoding error vs. the number of pixels per patch.

As described in [3], the probability of decoding error for B bits, assuming the decoding errors for the individual colors in the barcode are statistically independent events, is:

$$P_B(B, p) = 1 - (1 - P(p))^n \quad (7)$$

$$n = \lceil B / \log_2 N \rceil \quad (8)$$

The probability of decoding error increases with the number of bits. Figure 2 (left) shows the probability of decoding error of $n = 81$ color patches (using $N = 3$) that represents $B = 128$ bits (which represents an Internet Protocol Version 6 address (IPv6)) vs. the approximate camera-barcode distance. The approximate camera-barcode distance can be determined by the assumption that the ratio of the barcode height on the sensor and the real barcode height is the same as the ratio of the focal length and the distance to the barcode. Therefore, the approximate camera-barcode distance can be formulated as

$$d = f \frac{bH}{sG} \quad (9)$$

where f is the focal length of the camera, b is the physical height of the barcode, s is the sensor height, G is the barcode height in the image in pixels, and H is the image height in pixels.

To put this result in context, the minimum number of pixels per patch to decode robustly is 81 in minimum distance method and 49 in our method when the barcode is captured from a fronto-parallel view with correct focus. This indicates that a color barcode system that uses our decoding approach requires 40% less space to encode 128 bits using only $N = 3$ colors. A color barcode system that uses the minimum distance method with $N = 7$ reference colors can encode 128 bits in the same space as our decoding method. Note that the color palette of $N = 7$ colors occupies more space on a barcode and cannot be used to encode information.

In more difficult situation, where the barcode image is captured with incorrect focus and/or from a viewing angle ranging $\pm 30^\circ$, a robust decoding can be achieved with 90 pixels per patch using minimum distance method and 72 pixels per patch using our method. In this case, our method requires 20% less space to encode 128 bits. The minimum distance method using $N = 4$ reference colors could encode 128 bits in the same space as our decoding method.

6. CONCLUSIONS

We have presented a new algorithm for color barcode decoding that can handle blurry barcode images as well as images that were taken from relatively long distances using a smart camera phone. Our iterative algorithm decodes the colors of all patches across the barcode by minimizing the overall observation error. We showed a higher data density using small number of colors compared with minimum distance method.

There are potential improvement and extension in our method. A more efficient search strategy in the space of $\mathbf{w}$ may make our algorithm faster. Extension to non-uniform blur across the barcode image could improve the decoding performance for barcode images under perspective projection or barcodes printed on a non-planar surface.

7. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IIS - 0835645.

8. REFERENCES

- [1] 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.
- [2] Fan Wang and Roberto Manduchi, "Color-constant information embedding," in *Trends and Topics in Computer Vision*, pp. 13–26. Springer, 2012.
- [3] 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.
- [4] Homayoun Bagherinia and Roberto Manduchi, "High information rate and efficient color barcode decoding," in *Computer Vision—ECCV 2012. Workshops and Demonstrations*. Springer, 2012, pp. 482–491.
- [5] Henryk Blasinski, Orhan Bulan, and Gaurav Sharma, "Per-colorant-channel color barcodes for mobile applications: An interference cancellation framework," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1498–1511, 2013.
- [6] 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.
- [7] 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.