

Mimetic Code Using Successive Additive Color Mixture

Shigeyuki KOMURO[†], *Nonmember*, Shigeru KURIYAMA^{†a)}, *Senior Member*, and Takao JINNO^{†b)}, *Member*

SUMMARY Multimedia contents can be enriched by introducing navigation with image codes readable by camera-mounted mobile devices such as smartphones. Data hiding technologies were utilized for embedding such codes to make their appearances inconspicuous, which can reduce esthetic damage on visual media. This article proposes a method of embedding two-dimensional codes into images based on successive color mixture for a blue-color channel. This technology can make the color of codes mimic those used on a cover image, while preserving their readability for current general purpose image sensors.

key words: image data hiding, QR code, additive color mixture, color variation, digital signage

1. Introduction

The recent trend of cross-media framework bridges different types of media by connecting their information in an internet space with a handy code representing relevant URLs. Two-dimensional barcodes are widely used for such a purpose and some methods eliminate the visual clutter of the code, for example, by introducing logo-styles [1], colors [2], or by making it invisible with special materials illuminated only with laser [3] or black light [4]. Although these codes are often printed in advertising papers, the spread of digital signage systems enables them to be embedded onto a display that can handle dynamic images.

The data-hiding methodology for dynamic images was introduced for watermarking digital cinema [5]. This method requires reference images and a long capturing session, which is unsuitable for interactive detections of information. Unobtrusive code [6] can capture an AR-marker displayed at high frequency by utilizing a rolling shutter mechanism of CMOS sensors. This method requires stable and precise control of the display so as to synchronize it with sensors, and the amount of transmittable data is insufficient for transmitting URLs.

This article proposes a new type of quick response code (i.e., QR-code) displayed on a monitor such as a digital signage system. This code can mimic the colors of a cover image or movie so that the appearance of embedded codes is invisible when observed from distant positions but becomes visible from near positions. We introduce herein a color variation of QR-code along with the time for obtain-

ing such effects.

2. Perceptual Color Variation

The human vision system recognizes two or more colors as one color when they are alternately displayed in high frequency, and such an effect is called *successive additive color mixture*. On the other hand, the image sensor can only detect such time variations if its shutter speed is high enough to capture the changes of colors without accumulating them. In addition, modern liquid crystal monitors can update images at about 60 frames per second (fps), which is fast enough to cause the effect of additive color mixture. Therefore, we alternately change colors of QR-code so that the recognized colors coincide with a cover image.

Such color variation with the monitor, however, often cannot completely generate the expected mixed color due to the instability of fps in run time. For this reason, two colors should be carefully selected to reduce the flicker caused by such irregularity. The cone cells of the human vision system react to the wavelength around red and green colors in a wider distribution than those around blue. This property causes less sensitivity to the variations of a blue component, and has been successfully utilized in unobtrusive codes [6], [7]. We therefore selected a pair of mixed colors by changing a blue component, as shown in Fig. 1.

The human vision system is also less sensitive to variations of hue than to those of brightness, and we also investigated the perceptual sensitivity of additive color mixture by selecting a pair of colors on a chromaticity coordinate; two components of the xyY colorimetric system were chosen so that their mean color coincided with the color of a cover image. Consequently, we experimentally found that some pairs cannot generate the expected results of color mixture, and that such a hue-based mixture is unsuitable for reliably



Fig. 1 Example of blue color variations.

Manuscript received March 31, 2014.

Manuscript revised July 22, 2014.

[†]The authors are with Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.

a) E-mail: sk@tut.jp

b) E-mail: jinno@cs.tut.ac.jp

DOI: 10.1587/transinf.2014MUL0002

obtaining desirable mimetic effects.

3. Method of Data Transmission

3.1 Encoding

Given a cover image whose pixel value at location \mathbf{p} is denoted by $\mathbf{I}(\mathbf{p}) = \{r(\mathbf{p}), g(\mathbf{p}), b(\mathbf{p})\}$ in a RGB color space, the two types of images, denoted by $\mathbf{I}^+(\mathbf{p})$ and $\mathbf{I}^-(\mathbf{p})$, are alternately displayed as

$$\mathbf{I}^\pm(\mathbf{p}) = \begin{cases} \{r(\mathbf{p}), g(\mathbf{p}), b(\mathbf{p}) \pm \delta\}, & \mathbf{p} \in C \\ \{r(\mathbf{p}), g(\mathbf{p}), b(\mathbf{p})\}, & \text{others} \end{cases} \quad (1)$$

where C represents the set of pixel locations at which the colors of QR code are set to 0 (black), and a double-sign (\pm) corresponds.

3.2 Offset Variation

The variation of the blue component by $\pm\delta$ is impracticable if the blue component of a cover image is near the upper or lower bounds. We found that this saturation of color variation degrades the decoding performance, and the amount of variation is therefore offset to $\pm\hat{\delta}$ as

$$\pm\hat{\delta} = \begin{cases} \pm\delta + \alpha, & b(\mathbf{p}) < \delta \\ \pm\delta - \alpha, & b(\mathbf{p}) > 255 - \delta \\ \pm\delta, & \text{others} \end{cases} \quad (2)$$

where α denotes the offset and a double-sign \pm corresponds.

3.3 Decoding

Figure 2 shows the flowchart of our decoding. The variance of RGB colors, denoted by $\{\tilde{r}(\mathbf{p}), \tilde{g}(\mathbf{p}), \tilde{b}(\mathbf{p})\}$, is computed at every pixel of an image. The 10 frames of images are successively sampled from a captured movie, which takes about half second. The estimate of blue variations, denoted by $E_{\tilde{b}}(\mathbf{p})$, is then obtained with a Gaussian filter for reducing noisy variations as:

$$E_{\tilde{b}}(\mathbf{p}) = \exp\left(\frac{-\mu^2}{2\sigma^2}\right) \tilde{b}(\mathbf{p}), \quad \mu = \max\left(\frac{\tilde{r}(\mathbf{p})}{\tilde{b}(\mathbf{p})}, \frac{\tilde{g}(\mathbf{p})}{\tilde{b}(\mathbf{p})}\right) \quad (3)$$

where we experimentally set $\sigma = 0.5$. This filter can separately detect the variation of a blue component alone, by canceling the variations at the pixels where the red and green components also vary. This property is effective in detecting codes for dynamic images (or movies).

Additionally, the values of $E_{\tilde{b}}(\mathbf{p})$ are further filtered with γ -curve to widen their distributions as

$$\hat{E}_{\tilde{b}}(\mathbf{p}) = 255 \left(\frac{E_{\tilde{b}}(\mathbf{p})}{255} \right)^{1/\gamma}, \quad (4)$$

where we experimentally set as $\gamma = 2$.

The output $\hat{E}_{\tilde{b}}$ is sent to a freely available decoder of

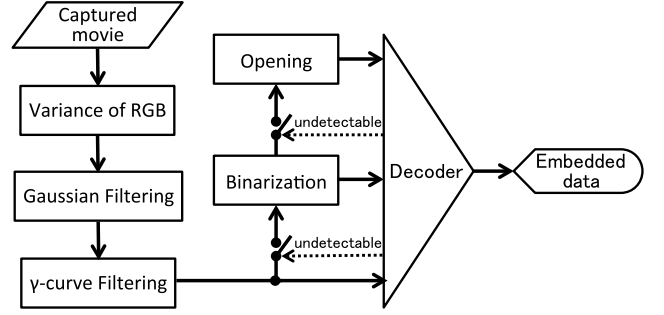


Fig. 2 Flowchart of code detection.

Table 1 Average computation times (sec.) for every processes.

Image size	QVGA	VGA
Movie capturing	0.5	0.5
Decoding & Validation	0.02	0.04
Gaussian & γ -curve Filtering	2.0	8.0
Binarization	0.3	0.8
Opening	2.0	7.0

QR-code[†], and is further processed on two stages; binarization based on discriminant analyses [8] and hole-filling with an opening operator of morphology, if the decoder returns an undetectable status after each stage, as shown in Fig. 2.

The average computation times for every processes are shown in Table 1. The time spent in decoding and validation is negligible compared to other processes, and thus no time is wasted at the two conditional branches. The first branch, however, could be omitted, because the frequency of undetectable responses from the decoder is very high. On the other hand, the second branch is necessary for avoiding wasteful computation spent in the opening process.

4. Evaluation of Decoding Performance

4.1 Experimental Conditions

We embedded QR-code of version 1, whose 30% of data can be used for error correction and 10 bytes are embeddable, which is sufficient for representing a shortened URL. We used a Nexus7 smartphone (Android 4.4, 1.5 GHz CPU) with a camera resolution of 5 mega pixels and frame rate of 24 fps. The camera exposure was set to minimum (-12) and the functions of auto-focus and white-balance were activated.

A movie was taken at the distance of 25cm from a 19-inch sized liquid crystal display of 60 fps (Diamondcrysta RDT196S). This distance is determined to be the farthest one for perfectly decoding QR-code with a monochromatic gray cover image. The QR-code was drawn at $10 \times 10 \text{ cm}^2$ physical size, and we captured movies at QVGA size (320×240 pixels) to reduce the computational cost for processing images.

[†]<http://zxingnet.codeplex.com>



Fig. 3 Sample cover images.

Table 2 Correct decoding rate for static cover images.

Variation ($= \delta$)	32	48	64	80	96
Lena	1.0	1.0	1.0	1.0	1.0
Country [†]	0.0	0.8	1.0	1.0	1.0
Sea ^{††}	1.0	1.0	1.0	1.0	1.0
Illustration	0.0	0.0	0.1	0.9	1.0
Road ^{†††}	0.8	1.0	1.0	1.0	1.0
Gradation	0.0	0.0	0.0	0.0	0.0

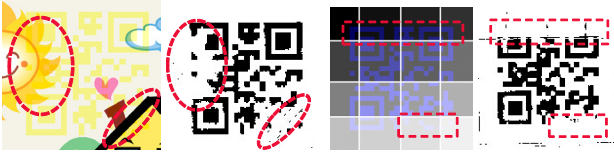


Fig. 4 Failures in detecting corner symbols.

4.2 Static Cover Images

Six types of static images, as shown in Fig. 3, are given as cover images. The images of (a) *Lena*, (b) *Country*, and (c) *Sea* were selected to have dominant color components of red, green, and blue, respectively. The images of (d) *Illustration* and (e) *Road* were selected to have globally strong and weak brightness, respectively, and (f) *Gradation* was synthesized to have all levels of brightness.

Table 2 shows the rate of correct decoding for 10 trials for each cover image, without using offset variation. As this result shows, the rate of correct decoding monotonically increases according to the increase of variations, and the rates for cover images of *Illustration* and *Gradation* are impractical. This failure was caused by the color of these cover images whose blue components reside near the upper or lower bounds.

Figure 4 demonstrates the defects in detecting corner symbols where the blue variations are insufficiently detected, owing to saturations of the blue. Table 3 shows

Table 3 Correct decoding rate with blue offset for $\delta = 32$.

Offset ($= \alpha$)	0	32	48	64	80	96
Country	0.0	0.6	0.8	1.0	1.0	1.0
Illustration	0.0	0.0	0.0	0.0	0.4	1.0
Gradation	0.0	0.0	0.0	0.0	0.3	0.8



Fig. 5 Sample cover movies.

Table 4 Correct decoding rate for cover movies.

Modulation ($= \delta$)	32	48	64	80	96
NightView ^{††††}	1.0	1.0	1.0	1.0	1.0
Beach ^{†††††}	0.0	0.0	0.1	0.4	0.7
Animal ^{††††††}	0.0	0.0	0.2	0.3	0.5

the improvement after applying offset variation $\hat{\delta}$ in Eq. (2), where the hardest condition of the variation $\delta = 32$ in Table 2 is employed.

We investigated about the effect of image size by changing the distances from display, and experimentally found that the correct decoding rate only depends on the density of pixels within the area of a QR-code. The number of pixels per each cell of the QR-code requires 5×5 pixels at minimum for ensuring stable decoding, which roughly corresponds to the distances of 25cm and 50cm for QVGA and VGA (640×480 pixels), respectively, for the conditions described in Sect. 4.1. The processing times were 4.8 and 16.3 seconds for QVGA and VGA, respectively, which approximately proportional to the number of pixels.

4.3 Dynamic Cover Movies

We next evaluated the decoding performance for the cover images with a dynamically changed appearance. Figure 5 shows three types of cover movies, and Table 4 shows the rate of correct decoding for 10 trials, where we adopted no offset $\alpha = 0$ for all movies.

The scene of *NightView* was slowly and slightly changed, and the decoding rate was practical. However, the scenes of *Beach* and *Animal* rapidly and greatly changed, causing difficulty in decoding even for large variations. The background of *Beach* contains the sea whose dominant color is blue, which often conflicts with the component of controlled variations for embedding data. However, Table 4 shows that this negative effect was not confirmed, because the performance of *Beach* has no significant difference against that of *Animal*.

Figure 6 shows the detected pattern of QR-code for

[†]<http://choppa24.biz/wp/beach/>

^{††}<http://www.nucba.ac.jp/press/entry-9683.html>

^{†††}<http://karukantimes.com/archives/51362848.html>

^{††††}<http://vimeo.com/22439234>

^{†††††}Clipped from a sample movie of Windows Media Player

^{††††††}Clipped from a sample movie of Windows Media Player

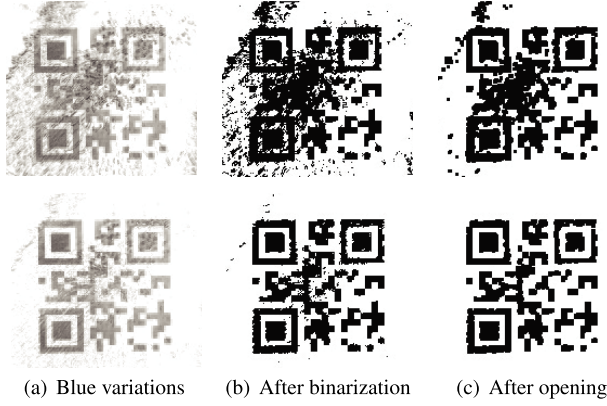


Fig. 6 Effect of blue-variation filter for animal movie.

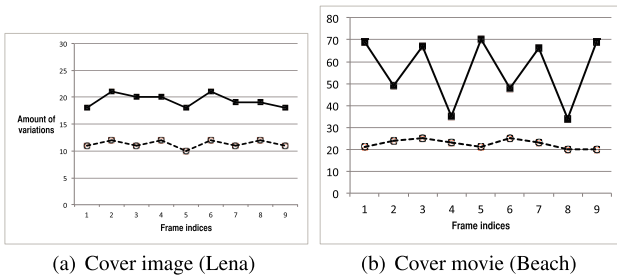


Fig. 7 Time variation of blue components at the cells of QR-code.

the cover movie of *Animal*, where upper images were taken without the filter of blue variations in Eq. (3), and lower ones were taken with the filter. This demonstrates the effectiveness of our Gaussian filtering for dynamic images, as explained in Sect. 3.3.

Figure 7 shows the variations of blue components along the sequence of frames for the cover image of (a) *Lena* and the movie of (b) *Beach*. Thick lines indicate the amounts of variations averaged among the correctly decoded areas inside black-colored squares of our QR-code. Dotted lines indicate those averaged among the incorrectly decoded (false positive) areas outside the squares of the code. This result shows that the variations are stably detected for the cover image, but are unstable for the movie. The dotted lines reveal the leak of variations to the surrounding areas, which often causes mis-detection of the coded symbols. Since the two variations of thick and dotted lines are separable, a perfect decoding might be attainable by introducing a smarter binarization.

Since the time required for capturing a movie is only about half second, the decoding process is expected to be robust against a camera vibration. We actually compared the correct decoding rate between the conditions of a camera fixed by a stand and held by both hands, and found that the decline in quality derived from the hand-held condition is negligible.

5. Evaluation of Mimetic Performance

Our mimetic QR-code is designed so that its appearance is

Table 5 Average and deviation of recognizable distances in meters.

Size of code	105×105	210×210	420×420
Lena	$0.67 (\pm 0.66)$	$1.3 (\pm 1.2)$	$4.6 (\pm 0.73)$
Illustration	$0.22 (\pm 0.51)$	$1.0 (\pm 1.1)$	$3.3 (\pm 1.6)$
NightView	$1.0 (\pm 0.87)$	$2.1 (\pm 1.2)$	$1.5 (\pm 1.1)$

only recognized when observed at near distances, and be invisible at far distances. We here confirm that property via a psychophysical experiment.

We asked 9 subjects, aged 22 to 45 and consisting of 8 men and 1 woman, if the displayed image includes our squared mimetic QR-codes. They were displayed at 105^2 , 210^2 , and 420^2 pixel sizes on a 50-inch LCD monitor, whose area was large enough to be used as a digital signage system, and all samples were synthesized without using offset ($\alpha = 0$). All subjects gradually changed their positions so that their distances from the monitor reached to $(5 - 0.5s)$, $s = 0, 1, 2, \dots, 9$ meters, and told whether they could recognize the existence of QR-code after displaying it for 3 seconds at a random location. The vertical span of the LCD monitor was 90 - 160 cm from a floor, which display the mimetic codes at about the same level of subjects' eyes for all standing positions.

Table 5 shows the average (and the standard deviation within parentheses) of the distances at which the subjects could recognize the code for the minimum variations of $\delta = 32$. Notice that we regarded the distance as 0 when the code is unrecognizable even at the minimum distance ($= 0.5$ meter).

The recognizable distances monotonically increased with respect to the size of codes, except that *NightView* had the maximum distance at 210^2 pixel sizes. The standard deviation estimates the individual variations in code discrimination. We also investigated the distances for the maximum variations $\delta = 96$, and found that more than half of the subjects recognized the code at the maximum distance ($= 5$ m) for every size, except that the *Illustration* of the smallest size ($= 105^2$ pixels) had the average distance of 3.6 meters. This shows that the color variations should be set around the minimum ($\delta = 32$) value to assure the practical mimetic property.

6. Conclusion and Discussion

We have investigated the feasibility of a new type of data-hiding technique using the additive color mixture of the human vision system. Practical decoding accuracy was obtained for static cover images; the decoding accuracy, however, is still impractical for images with scenes that greatly and rapidly change.

The limitation of our method is that the visual quality is insufficient for simple and artistic images whose blue components reside near the upper or lower bounds, due to the influence of supplemental offset. Another color-hiding approach should be therefore developed for diminishing the demerit caused by the offset. The estimation of suitable regions for drawing our mimetic code is also an important

topic for enhancing its usability.

References

- [1] LogoQ, <http://logoq.net/logoq/logoq/index.html>
 - [2] Color Code, SmartIcon, <http://www.colorzip.co.jp/>
 - [3] J.M. Meruga, W.M. Cross, M.P. Stanley, Q. Luu, G.A. Crawford, and J.J. Kellar, "Security printing of covert quick response codes using upconverting nanoparticle inks," *Nanotechnology*, vol.23, no.39, pp.395201, 2012.
 - [4] K. Kamijo, N. Kamijo, and M. Sakamoto, "Electronic clipping system with invisible barcodes," *Proc. 14th Annual ACM International Conference on Multimedia*, pp.753–762, 2006.
 - [5] J. Haitisma and T. Kalker, "A watermarking scheme for digital cinema," *Proc. ICIP*, vol.2, pp.487–489, 2001.
 - [6] G. Woo, A. Lippman, and R. Raskar, "VRCodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter," *Proc. 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp.59–64, 2012.
 - [7] FPcode, <http://jp.fujitsu.com/solutions/fpcode/>
 - [8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man. Cybern.*, vol.9, no.1, pp.62–66, 1979.
-