Be-code: Information Embedding for Logo Images

Kazuha Yamada
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi
400-8511, JAPAN

Yasuna Yamashita
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi
400-8511, JAPAN

Masahiro Toyoura
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi
400-8511, JAPAN

Xiaoyang Mao
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi
400-8511, JAPAN

Satoshi Takatsu
CRESCO Co., Ltd.
2-15-1 Konan, Minato-ku, Tokyo
108-6026, JAPAN

Chihiro Sugawara
University of Yamanashi
4-3-11 Takeda, Kofu, Yamanashi
400-8511, JAPAN

ABSTRACT

2D image codes are widely used for inputting information to mobile devices with cameras. QR code is a typical example of such codes. Conventional image codes do not have semantics in images themselves. Such codes may spoil the quality of design when attached on commercial products. We propose a novel code, Be-code, for embedding data into logo images. Be-code is generated from a logo image by modifying some of the edge pixels according to the bits to be embedded, but avoiding causing obvious change to the appearance of the original logo image. Algorithm for extracting information from the camera captured Be-code is also presented. The original logo image is not required in decoding. In experiments, embedded data could be decoded at a high success rate.

Categories and Subject Descriptors

I.3.3 [**Picture/Image Generation**]: [Display algorithms]; I.4.1 [**Digitization and Image Capture**]: [Scanning]

General Terms

Theory

Keywords

Image code, data embedding, flippability score, homography

1. INTRODUCTION

2D image codes such as QR code [6] and MicrosoftTag [3] are widely used as an effective way to bridge real world and digital world. With the popularization of camera installed mobile devices, we can find these codes everywhere, from product packages, advertising circular to airline tickets. People use them to connect their customers from the offline marketing materials to online information, entertainment, and interactive experiences. However, these conventional codes do not have semantics in images themselves. Most of them are simply 2D bar codes and hence may spoil the design when being attached to commercial products.

In this paper, we propose a new 2D code called Be-code (Binary emblem code). Be-code is generated from a logo image. The appearance of the original logo remains in the generated Be-code. Since a logo is the symbol of a brand identity, enabling an easy link from a logo image to the online information would benefit both consumers and suppliers in various applications. However, the design of Be-code faces two challenges: First, most logos are abstracted symbols and there is little information redundancy can be used for embedding information, whereas noticeable changes to the original image would ruin the brand identity. Second, we should allow easy decoding with mobile device under various capturing conditions. Again for avoiding ruining the design, we cannot place markers into logos, which makes it difficult to estimate locally adapted parameters for reconstruction the original image from a captured one. We solve the first problem by extending the data hiding technology proposed by Wu *et al.* [11]. Wu *et al.*'s method embeds data into binary images through flipping particular pixels at boundaries. We propose a new technique to partially preserve the antialiasing effect at those boundaries. To approach the second issue, we have developed a new technique to estimate the homography between the captured image and the original coordinates using four least noticeable markers placed out of the logo. Experimental results show that Be-code could be decoded at a high success rate with mobile devices under casual capturing conditions.

The remainder of the paper is organized as follows: after a briefly review of related works in Section 2, section 3 gives the overview of Be-code and its encoding/decoding procedure. Section 4 introduces the details of the proposed algorithms. Section 5 describes the results of experiments and Section 6 concludes the paper.

2. RELATED WORKS

QR code [6] represents data in black and white pattern. Many existing work have been done to improve the design of QR code. Design QR [5] and QR-JAM [4] reserve some area in QR code for including a small color image by making use of the redundancy of the code. ColorCode [2] and MicrosoftTag [3] use color patterns instead of the black and white pattern. Although those enhanced codes are visually more pleasing than the original QR code, they all suffer from the lacking of design flexibility, since they are basically 2D bar codes. FPcode [10] and CLIC2C [1] uses water marking technology to embed information into images. However, these techniques embed information by altering the intensity or color of original images and hence are not suitable for many logo images, which are abstract symbols with no locally change of color or intensity.

Miyake *et al.* [7] proposed a technique for encoding data into an illustration by attaching a bold line to the contour of the illustration. Data are encoded as 1D color pattern in bold line. However, the attached bold line may ruin the design of the logo in most cases. Pramila *et al.* developed a method for embedding information into a binary image by adding a solid rectangle frame to the image [9]. Their method assumes a fixed size for the image, which inherently restricts its usefulness in real applications.

Our technology is built upon the existing binary image watermarking technique proposed by Wu *et al.* [11]. They introduced the notion of flippable pixels which are the pixels that can be inverted (flipped) without noticeable corruption of original image, together with a method for computing the flippability score by considering the smoothness and connectivity of binary images. Although Zubarev *et al.* [12] improved the embedding capacity and the time efficiency for finding the flappable pixels by employing the wet paper coding technology, we chose [11] for avoiding noticeable corruption of original images as well. While most watermarking techniques assume the decoding from digital images, Be-code is mainly used for linking the real world to online. Therefore, our major contribution is the new encoding/decoding methods allowing the extraction of information from the images captured with mobile devices under casual capturing conditions.

3. OVERVIEW OF BE-CODE

3.1 Configuration

The structure of Be-code is shown in Figure 1. The logo is placed in an emblem area with data embedded in its contour. A direction marker and four frame markers are placed out of the emblem area. The direction marker indicates the left-top of the Be-code. The frame markers are used for defining the boundary of the Be-code, as well as for estimating homography required in reconstruct the original image from the camera captured one. Margins are inserted for separating the logo from markers and for separating Be-code from the other contents.

3.2 Framework of Encoding/Decoding

The overall procedure of Be-code generation and decoding is shown in Figure 2. A bit string is embedded into a logo image and the frame markers are attached. Printed Be-code

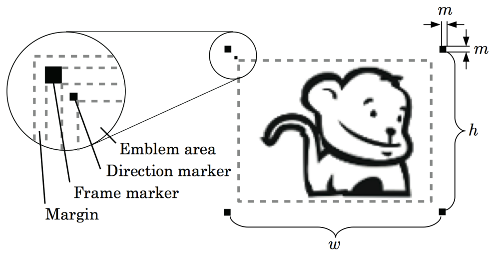


Figure 1: Structure of Be-code.

is captured by a camera of mobile device. A homography is estimated based on the position of frame markers. The Be-code in the original coordinates is reconstructed with the homography matrix. The bit string is decoded from the reconstructed Be-code.

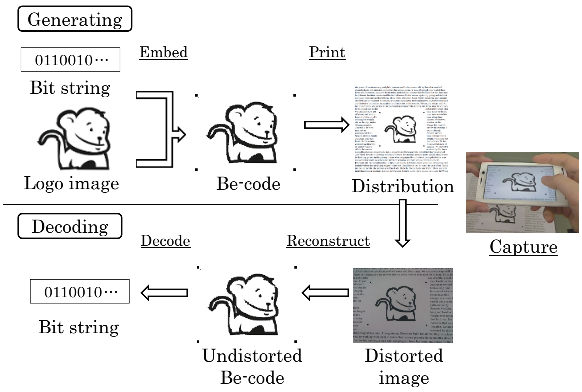


Figure 2: Overall procedure for generating and decoding of Be-code.

Reed-Solomon coding is applied to the embedded data. A small number of errors can be corrected when capturing the Be-code. Reed-Solomon coding also helps to validate the decoding.

4. REALIZATION OF BE-CODE

4.1 Reconstruction from Captured Image

To reconstruct the original Be-code from a captured image, we use the frame markers. A homography between captured image and the original coordinates is estimated with the relative positions of frame markers in the captured image. One problem is that the size of the original image is unknown. We solve the problem by using a two-step mapping algorithm with the given condition that the size of frame markers is $m \times m$ pixels. At the first step, we use the fact that Be-code is a rectangle to compute a preliminary homography. If we assume the size of the original Be-code be $w' \times h'$, the positions of four frame markers in the captured image give a homography. Reconstructed Be-code has correct ratio of marker size to Be-code size in both width and height, though the correctness of the sizes themselves is not guaranteed. With the condition that the size of frame markers is $m \times m$ pixels, the correct width w and height h of Be-code can be calculated with the following equations:

$$w = m\, w'/m'_x \quad (1)$$

$$h = m\, h'/m'_y \quad (2)$$

Here, m'_x and m'_y are the width and height of the maker in the preliminary reconstructed image. With the correct size of the original Be-code ready, the second step reconstructs each pixel of the original Be-code with the homography.

4.2 Data embedding with flippability score

Since most logos are abstract symbols and have little redundancy in colors and intensity, we embed the data to the contour of the symbol by extending the flippability score based binary image data hiding technique proposed by Wu *et al.* [11]. As shown in Figure 3, a binary image is segmented into 3×3 blocks and a flippability score is calculated for each block. The flippability score gives a quantitative measure to how unnoticeable the change is when the center pixel of the block is flipped (please refer to [11] for the details of flippability score calculation). A bit string is embedded into the binary image by flipping the center pixel of the block with a flippability score higher than 0.5. That is, to embed a "0", we change the center pixel to black, and to embed a "1". we flip the center pixel to white. The most plausible property of the flippability score is that a block with a score above 0.5 will retain its score to be above 0.5 after its center pixel is flipped. This property enables us to extract the embedded data from the reconstructed Be-code by finding out the blocks with a score above 0.5, without referring to the original image.

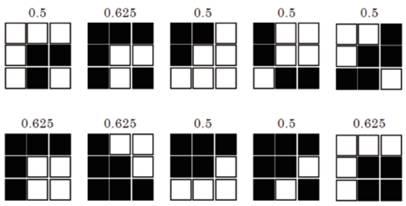


Figure 3: Examples of flippablility scores of 3×3 blocks.

Wu *et al*'s original method segments an image into non-overlapping 3×3 blocks. We segment an image into blocks with one pixel line overlapped as shown in Figure 4. While the original method takes 1/9 pixels of the whole image as the candidate for flipping, our proposed segmentation can make use of 1/4 pixels of the whole image, and hence has a larger embedding capacity than the original method.

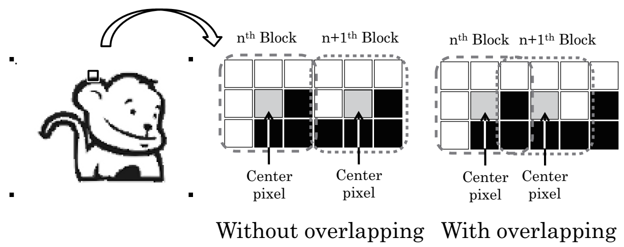


Figure 4: Increasing the embedding capacity by overlapping blocks.

4.3 Preserving of Anti-Aliasing

Although a logo image usually has flat color or intensity, the pixels at the boundary of the symbol usually have grayscale values due to anti-aliasing. It is important to preserve those grayscale values for keeping the visual quality of the logo. We achieve this by performing the binarization on the fly only for calculating the flippability score and for extracting the embedded bits while leaving the grayscale values unchanged in the Be-code. However, the grayscale values of the captured Be-code can be largely affected by the illumination condition. To assure a successful binarization, we add (or subtract) an offset to (or from) the gray level of the pixels which should be converted to white (or black) with the binarization, to keep a safety margin between them.

As shown in Fig 5, from the original grayscale image, the binarization threshold g_{th} is calculated with Otsu's method [8]. The pixels with a gray level near g_{th} may flip their values when the captured image is binarized. To avoid the flipping, we push the pixels toward the black and white end with the offsets $b_{th} = g_{th} - c\, g_{th}$ and $w_{th} = g_{th} + c(255 - g_{th})$, respectively, in the following way:

1. if $b_{th} < v < g_{th}$, $v = b_{th}$.
2. if $g_{th} \leq v < w_{th}$, $v = w_{th}$.
3. if else, v is not changed.

Here, c $(0 < c < 1)$ is a user given constant for trading the visual quality with the robustness of decoding. A smaller c remains the original grayscale more, and a larger c makes decoding robust even when the illumination changes.

Figures 6(a) and 6(c) are an example of binary Be-code, and Figures 6(b) and 6(d) are the Be-code generated with the proposed method. We can see that the anti-aliasing effects are partially preserved in the Be-code of Figure 6(b) and 6(d).

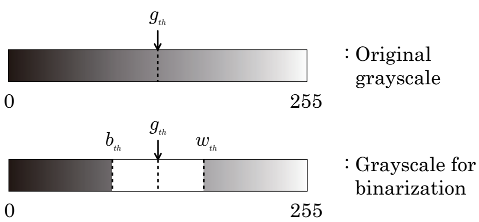


Figure 5: Conversion of grayscale considering binarization.

5. EXPERIMENT

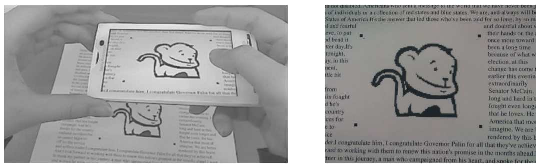
We have conducted experiments to investigate the capacity of embedding, the success rate of decoding and the visual quality of Be-code for a variety of logo images. In the experiment, Be-code is captured and decoded with Xperia(SO-01B) made by Sony Ericsson as shown in Figure 7.

Figure 8 shows several examples of generated Be-codes together with the amounts of embedded data. The size of each original image is 110×80 pixels. Logos with relative complex shape, such as (a) and (c), have larger embedding capacity. The logo with diagonal lines, such as (b), also has a large capacity. By overlapping blocks for selecting flippable pixels, we succeeded in increase the capacity of embedding.

(a) Be-code with binary values. (b) Be-code with grayscale values.

(c) Zoom-up image of (a). (d) Zoom-up image of (b).

Figure 6: Preserving anti-aliasing in Be-code.



(a) Experimental environment. (b) An example of captured images.

Figure 7: Experimental environment for capturing Be-codes.

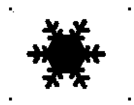
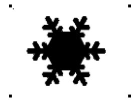
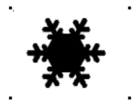
	(a) Monkey	(b) Letter	(c) Snow
Original images			
Be-code (c=1)			
Be-code (c=1/2)			
Be-code (c=1/3)			
Non-overlapping	58 bits	56 bits	45 bits
Over-lapping	145 bits	113 bits	86 bits

Figure 8: Be-codes and their embedding capacity.

Figure 8 also shows the results obtained by varying the parameter c for controlling the tradeoff between anti-aliasing effect and robustness. A smaller c retains the original grayscale more. A subject captured each Be-code 100 times under a usual illumination condition of our laboratory. The success percentage of decoding are shown in Table 1. Be-codes with $c = 1/2$ has a success rate closing to 100% for all images.

Table 1: Success rate of decoding Be-codes with $c = 1/3$ and $c = 1/2, 1$.

c	1/3	1/2	1
Monkey	34.0 %	98.0 %	100.0 %
Letter	91.0 %	100.0 %	99.0 %
Snow	51.0 %	100.0 %	100.0 %

6. CONCLUSIONS

We proposed Be-code as a new technique for linking logo images with online information. In experiments, embedded data could be decoded at a high success rate.

As a future work, we need to confirm the robustness of Be-code when captured outdoors or under some special illumination environments. Currently our algorithm can only be applied to grayscale logo images or the logo images consisting of a single color. Extension to logos with multiple colors is another important future work.

7. REFERENCES

[1] A. M. Alattar. Smart images using digimarc's watermarking technology. 3971(25):264–273, 2000.

[2] *ColorCode*, http://www.colorzip.com.

[3] *Microsoft Tag*, http://tag.microsoft.com/.

[4] M. Hagiwara. Decorated 2D codes. *The Journal of the Institute of Electronics, Information and Communication Engineers*, 94:341–343, 2011.

[5] *Design QR*, http://d-qr.net/.

[6] *QRcode*, http://www.qrcode.com/en/index.html.

[7] T. Miyake and S. Kuriyama. Data embedding by changing brightness on shape contours. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences (Japanese Edition)*, J93-A(2):73–81, 2010.

[8] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[9] A. Pramila, A. Keskinarkaus, and T. Seppaenen. Reading watermarks from printed binary images with a camera phone. 5703:227–240, 2009.

[10] G. Suzuki, N. Yamaguchi, S. Nakamura, and H. Chiba. Mobile interaction using steganographic image on mobile display. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 507–510, 2008.

[11] M. Wu and B. Liu. Data hiding in binary image for authentication and annotation. *IEEE Transactions on Multimedia*, 6(4):528–538, 2004.

[12] M. Zubarev, V. I. Korzhik, and G. Morales-Luna. Wet paper coding for watermarking of binary images. *CoRR abs/0808.2486*, 2008.