A New Data Transfer Method via Signal-rich-art Code Images Captured by Mobile Devices

Ya-Lin Lee, *Student Member, IEEE,* and Wen-Hsiang Tsai, *Senior Member, IEEE*

Abstract—**A new type of signal-rich-art image for applications of data transfer, called signal-rich-art code image, is proposed. The created code image is visually similar to a pre-selected target image and with a given message embedded, achieving the effect of the so-called signal rich art. With its function similar to that of a QR code, such a type of image is produced by encoding the message into a binary bit stream, representing the bits by binary code patterns of 2×2 blocks, and injecting the patterns into the target image by a novel image-block luminance modulation scheme. Each signal-rich-art code image may be printed or displayed, and then re-captured by a mobile-device camera. Skillful techniques for counting the number of pattern blocks and recognition of code patterns are also proposed for message extraction from the re-captured version of the signal-rich-art code image. Good experimental results and a comparison of them with those of an existing alternative method show the feasibility and superiority of the proposed new data transfer method.**

Index Terms—**Signal rich art, data transfer, barcode, QR code, signal-rich-art code image, code pattern.**

I. Introduction

Signal rich art, as defined by Davis [1], is the art that communicates its identity to context-aware devices, where "art" includes all forms of creative communication. Many types of identities existing in the human environment can be utilized as signal-rich-art carriers, such as digital media, pictorial material, artwork, etc. For the purpose of depicting or recording human visual perception results, the image/video identity has more artistic effects than other types of identities. In this paper, we define *signal-rich-art image* as the type of signal rich art with its identity being an image which can not only be any digital file but also be any real object, such as posters, labels, illustrations, etc. Signal-rich-art images can help people to conduct ubiquitous computing [2]; they can exchange information via such images existing in the environment everywhere and anytime. For example, one can use the camera on a smart phone to capture an image of an advisement on a magazine or a painting displayed in an exhibition to obtain the detailed information related to the advertisement or the painting.

Two common techniques for signal-rich-art communication are the use of barcodes and data hiding [1]. Barcodes, which are usually attached to objects for various identification purposes, represent machine-readable data by patterns of lines, rectangles, dots, etc. For example, Fig. 1 shows some commonly-used barcodes, including Code 39 [19], PDF417 [20], QR code [21], and data matrix code [22]. The data encoded into such barcodes can be extracted using barcode reading techniques [3]-[6]. For instance, Ouaviani *et al.* [3] proposed an image processing framework for 2D barcode reading, which includes four main phases: region-of-interest (ROI) detection, code localization, code segmentation, and decoding. Zhang *et al.* [4] proposed a real-time barcode localization method by a two-stage process, which segments first the barcode shape in a low-resolution image by region-based analysis, and then extracts the barcode meaning from the image of the original resolution. Yang *et al.* [5] proposed another accurate barcode localization method by using some prior knowledge of the barcode shape to detect corners initially, followed by more accurate corner localization. Yang *et al.* [6] proposed an adaptive thresholding technique for binarizing barcode images by constructing a dynamic search window centered at the edge pixel nearest to each pixel to be binarized.

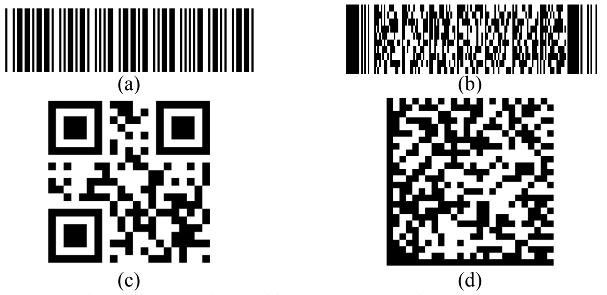


Fig. 1. Examples of commonly-used barcodes. (a) Code 39. (b) PDF 417. (c) QR code. (d) Data matrix code.

In addition to the use of barcodes, data hiding is an alternative signal-rich-art communication technique that embeds data into cover media for applications like covert communication, copyright protection, authentication, etc. With the advance of computer technology, many data hiding methods have been applied on *digital* cover media, such as images, videos, audios, text documents, etc. [7]-[11]. However,

Manuscript received July 20, 2013. This research was supported in part by the NSC, Taiwan under Grant No. 101-3113-P-009-006 and Grant No. 102-2218-E-009-003, and in part by the Ministry of Education, Taiwan under the 5-year Project of "Aiming for the Top University" from 2011 through 2015.

Y. L. Lee is with the Institute of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan 30010 (e-mail: yllee.cs98g@g2.nctu.edu.tw).

W. H. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 30010. He is also with the Department of Information Communication, Asia University, Taichung, Taiwan 41354 (email: whtsai@cis.nctu.edu.tw).

these data hiding methods transfer data via *digital* files only. Furthermore, they are mostly insufficient to enable the signal-rich-art effect when one wants to interact with the surrounding environment. Such methods may be called "*digital*" *data hiding*.

Another type of data hiding, which may be called "*hardcopy*" *data hiding*, can embed information into the so-called *image barcodes* using halftone techniques [12]-[14]. These barcodes have the visual appearances of other images and the encoded information can be decoded from their hardcopy versions acquired by *scanners*. That is, the encoded information can survive *"print-and-scan attacks."* For example, Bulan *et al.* [12] proposed a framework for data hiding in images printed with clustered dot halftones via a pattern orientation modulation technique. Bulan and Sharma [13] proposed another pattern orientation modulation technique that utilizes three printing channels and modulates the orientations of elliptical-shaped dots for data encoding. Damera-Venkata *et al.* [14] proposed a block-error diffusion method that embeds information into hardcopy images by using dot-shape modulation.

However, if one uses a mobile device to capture the image of a *hardcopy* of the above-mentioned image barcodes, the information might not be decoded successfully since the captured image will suffer from more types of distortions than those acquired by scanning, such as perspective deformation, noise addition, blurring, uneven lighting, etc. Recently, Lee and Tsai [15] proposed a new type of signal-rich-art image, which is called *signal-rich-art character image* here. To the best of our knowledge, this is the first work that can solve the above-mentioned issues. Specifically, a signal-rich-art character image is created from a target image used as a carrier of a given message by fragmenting the shapes of the composing *characters* of the message and "injecting" the resulting character fragments randomly into the target image by a block luminance modulation scheme. Each signal-rich-art character image so created has the visual appearance of the corresponding pre-selected target image while conventional barcodes do not. Also, the data embedded by Lee and Tsai's method [15] can be extracted from a "camera-captured" version of the created signal-rich-art character image while those embedded by the use of the aforementioned hardcopy data hiding methods using image barcodes cannot. The function may be implemented on a mobile device.

However, as shown in Fig. 2(b), each signal-rich-art character image generated by Lee and Tsai [15] contains many small character fragments with undesired visual effects. Also, it requires an optical character recognition scheme to extract the embedded message, which is usually time-consuming. Also, the size of each block cannot be too small in order to keep the resolution in the captured image sufficiently good for correct extraction of the character shapes in the image. To solve these problems, another new type of signal-rich-art image, called *signal-rich-art code image*, is proposed in this study. Specifically, instead of transforming the given message to be embedded into a character message image, the message is converted, in the sense of *data coding*, into a bit stream of codes first, which is then represented by binary *pattern blocks*, each being composed of 2×2 *unit blocks*. A block luminance modulation scheme is then applied to each pattern block to yield a signal-rich-art code image with the visual appearance of a pre-selected target image. An example of the resulting signal-rich-art code image is shown in Fig. 2(c), which is more pleasing than the signal-rich-art character image shown in Fig. 2(b) generated by Lee and Tsai [15]. A more detailed comparison with Lee and Tsai [15] by experiments reveals the following additional merits of the proposed method: (1) the yielded signal-rich-art code image has a much better visual appearance of the target image; (2) the accuracy rate of message extraction from the generated code image is higher; (3) the message extraction speed is higher.

In the remainder of this paper, the idea of the proposed method is described in Section 2. The details for signal-rich-art code image generation and message extraction are given in Sections 3 and 4, respectively. In Section 5, experimental results are presented to show the feasibility of the proposed method, followed by conclusions in Section 6.

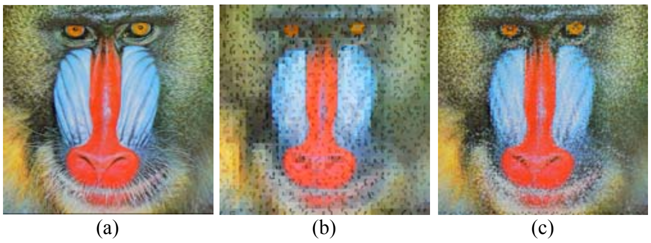

(a) (b) (c)

Fig. 2. Examples of signal-rich-art images yielded by Lee and Tsai [15] and proposed method. (a) Target image. (b) Signal-rich-art character image created by [15]. (c) Signal-rich-art code image created by proposed method.

II. Idea of Proposed Method

The proposed method includes two main phases of works as illustrated in Fig. 3: 1) signal-rich-art code image generation; and 2) message extraction. In the first phase, given a target image I_T and a message M, a signal-rich-art code image I_C is created by four major steps:

Step 1.1 — transform message M into a bit stream B of codes;

Step 1.2 — transform every three bits of B into four bits and represent them by a binary pattern block, resulting in a *pattern image* I_P;

Step 1.3 — modulate each pattern block T_i of I_P by two *representative values* calculated from the Y-channel values of the corresponding block B_i of target image I_T, yielding a *modulated* pattern image I_P';

Step 1.4 — replace the Y-channel of target image I_T with I_P' to get a signal-rich-art code image I_C as the output.

In the second phase, given a camera-captured version I_C' of a *paper or display copy* of the signal-rich-art code image I_C, a message M', which is supposed to be identical to M, is extracted from I_C' by four major steps:

Step 2.1 — localize the region I_C'' of the original part of the signal-rich-art code image I_C in I_C';

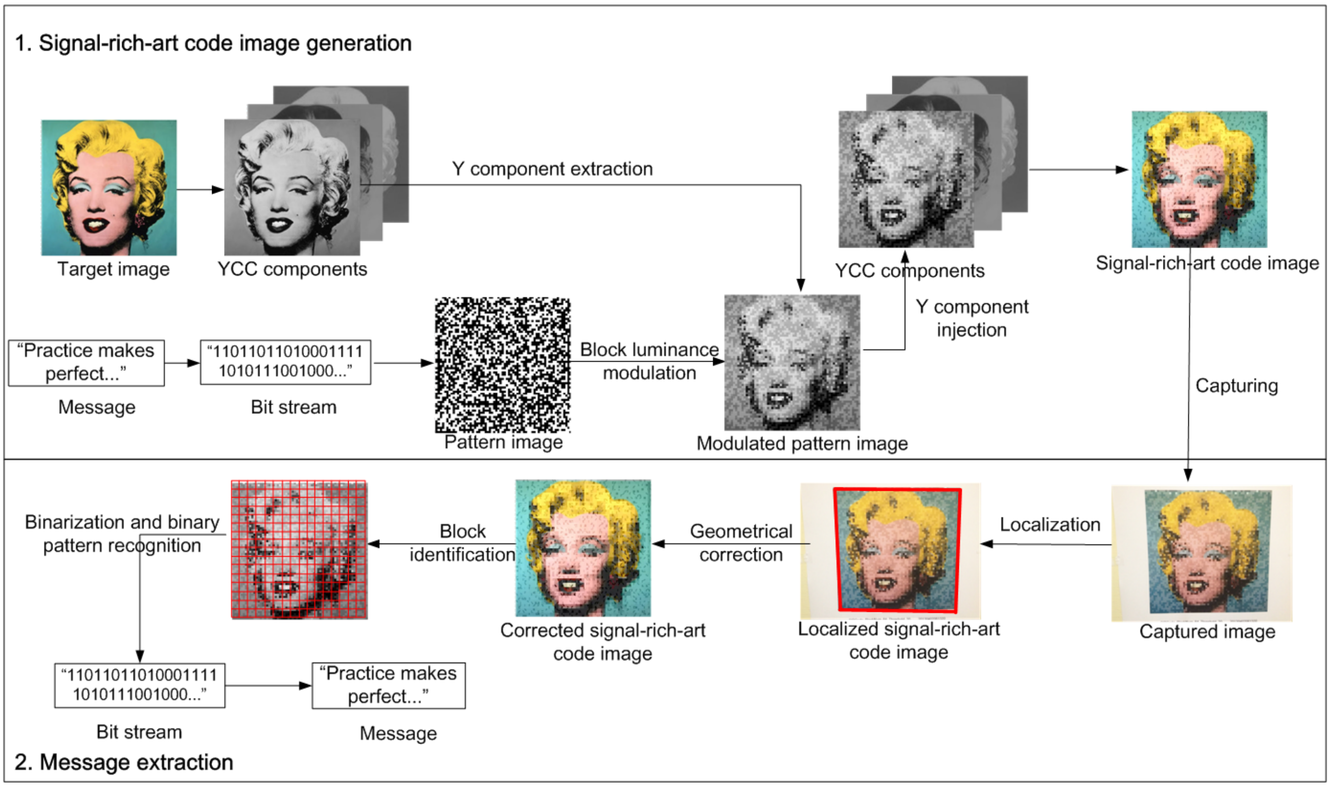


Fig. 3. Illustration of major steps of two phases of proposed method.

Step 2.2 —correct the geometric distortion in I_C'' incurred in the image acquisition process, yielding a corrected image I_C''';

Step 2.3 — identify the unit blocks in I_C''' automatically and divide I_C''' accordingly into pattern blocks, each with 2×2 unit blocks;

Step 2.4 — binarize each pattern block of I_C''', recognize the result to extract the bits embedded in it, compose all the extracted bits to form a bit stream B, and transform B reversely to get a message M'.

III. Generation of Signal-Rich-Art Code Image

A. Pattern Image Creation

Unlike Lee and Tsai [15] who transforms a message M into a *character message image*, the proposed method transforms M into a bit stream B of codes, uses binary *code patterns* to encode the bits of B, and composes the code patterns, each in the form a *pattern block*, to form a *pattern image* similar in appearance to a pre-selected target image. Specifically, each pattern block T consists of several unit blocks F_i, with each F_i representing a bit of the code pattern C which T represents. A main issue here is how to design the code patterns so that the corresponding pattern blocks are suitable for use not only in message embedding but in block luminance modulation (see Step 1.3 above). To solve this issue, two characteristics must be provided in the designed code patterns: 1) the number of bits in each code pattern C must be small enough, so that the pattern block T representative of C can keep the *local* color characteristic of the corresponding target image area; and 2) the colors of the unit blocks F_i of the pattern block T representative of each code pattern C should not be all the same, since otherwise the original bits represented by the unit blocks of the code patterns will become *undistinguishable* during the message extraction process.

The first characteristic mentioned above is necessary for the resulting signal-rich-art code image to become more similar to the pre-selected target image. And as an illustration of the necessity of the second characteristic, Fig. 4 shows an example of undistinguishable binary code patterns, where the unit blocks F_i of the pattern block T representative of a code pattern C with bits "0000" are all of an identical color originally and are modulated to be all of another color, but then in the message extraction stage, the bits represented by the modulated pattern block cannot be extracted since only one color exists in this modulated pattern block and the bits corresponding to this color cannot be uniquely determined (more details discussed later).

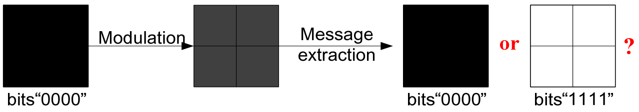


Fig. 4. An example of undistinguishable binary code patterns.

Therefore, in this study each pattern block representative of a code pattern is set to be of the smallest size of 2×2 unit blocks. Also, a *bit expansion* scheme of appending an odd parity bit to every three bits of the bit stream B is conducted, making the resulting four bits being not all the same and satisfying the above-mentioned second required characteristic of the code pattern. In detail, let the bit stream B be denoted as

$$B = b_{11}b_{12}b_{13}b_{21}b_{22}b_{23}b_{31}b_{32}b_{33} \ldots b_{n1}b_{n2}b_{n3};$$

and for every three consecutive bits $b_{i1}b_{i2}b_{i3}$ in B, we append an odd parity bit b_{i4}' to get four bits $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$ according to the following rule:

$$\textit{set } b_{i4}' = \overline{b_{i1} \vee b_{i2} \vee b_{i3}} \ \textit{ and } b_{ij}' = b_{ij} \text{ for } j = 1, 2, 3, \qquad (1)$$

where $\vee$ and $\overline{\cdot}$ denote bitwise "OR" and "complement" operations, respectively. The resulting four bits $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$ are not all identical, as can be verified by ORing them to get the following result:

$$b_{i1}'\vee' b_{i2}'\vee b_{i3}\vee b_{i4}' = (b_{i1}\vee b_{i2}\vee b_{i3})\vee \overline{b_{i1} \vee b_{i2} \vee b_{i3}} = 1, \qquad (2)$$

which means that at least one "1" must exist in $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$, and if all the four bits are "1s," then all the three bits b_{i1}', b_{i2}', and b_{i3}' must be 1's, leading to the result $b_{i4}' = \overline{b_{i1} \vee b_{i2} \vee b_{i3}} = 0$, which is a contradiction. Moreover, the total possible number of distinct expanded four bits $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$ for different combinations of $b_{i1}b_{i2}b_{i3}$ is eight, as shown in Fig. 5. These eight combinations are taken as the code patterns which we mentioned previously.

Next, we create a 2×2 pattern block $T_i = F_{i1}F_{i2}F_{i3}F_{i4}$ with four unit blocks F_{i1} through F_{i4} to represent the non-all-identical bits $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$ of each code pattern C_i, where the color of unit block F_{ij} is set to be black if the corresponding bit is 0; or to be white if the corresponding bit is 1. Accordingly, as can be seen from Fig. 5, the colors of the pattern blocks representative of the eight code patterns are all non-identical as well.

Message bits $b_{i1}b_{i2}b_{i3}$	*Code pattern* $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$	*Pattern block T_i*
000	0001	■■ ■□
001	0010	■■ □■
010	0100	■□ ■■
011	0111	■□ □□
100	1000	□■ ■■
101	1011	□■ □□
110	1101	□□ ■□
111	1110	□□ □■

Fig. 5. Performing bit expansion scheme on every three message bits to yield eight binary code patterns represented by pattern blocks.

Finally, we create a *pattern image* I_P of the size of the target image I_T by arranging all the pattern blocks T_i, say n ones, in a raster-scan order. If the n pattern blocks do not fill up I_P, then we repeat to fill them into I_P again and again until they do. For example, with the target image I_T as shown in Fig. 6(a) and the bit stream B = "110110110100011111010111001...," the pattern image I_P resulting from such filling operations is shown in Fig. 6(b).

B. Block Luminance Modulation

After the pattern image I_P is created, it is "injected" into the target image I_T under the constraint that the resulting image retains the visual appearance of I_T. For this, we utilize a characteristic of the YCbCr color model — the luminance component Y is *independent* of the others [16] — to embed I_P into the Y-channel of I_T. This will solve a problem of illumination variation encountered in the later stage of message extraction. A block luminance modulation technique is proposed for use here, which modulates the mean of each pattern block T_i to be the same as that of the corresponding target block B_i of I_T. The resulting modulated pattern image I_P' so has roughly the visual appearance of the Y-component of the target image I_T. For example, Fig. 6(d) shows a modulated pattern image I_P' so created, which looks like the Y-component of the target image I_T shown in Fig. 6(c); and Fig. 6(e) shows a zoom-out version of part of Fig. 6(d) enclosed by the red rectangle.

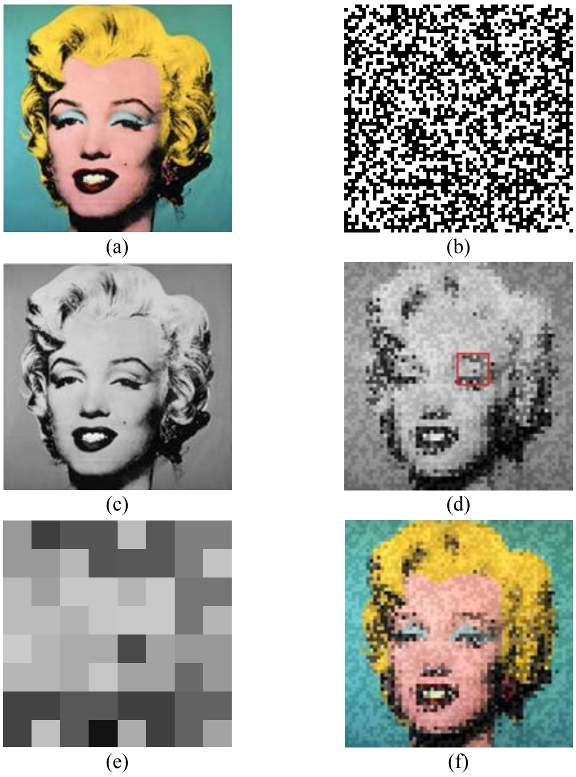

(a) (b) (c) (d) (e) (f)

Fig. 6. Signal-rich-art code image generation. (a) Target image. (b) Pattern image I_P. (c) Y-channel of (a). (d) Modulated pattern image. (e) Zoom-out of red square region in (d). (f) Resulting signal-rich-art code image.

More specifically, firstly the Y-component of the target image I_T is divided into blocks, denoted by B_i, all with the same size as that of the pattern blocks in the pattern image I_P. Then, let N_B and N_W denote the numbers of black and white pixels in T_i, respectively. The pixels of each B_i are sorted according to their Y values in an ascending order to obtain an ordered Y-value set $\{q_1', q_2', \ldots, q_m'\}$. Then, two representative Y values r_1 and r_2 are computed for B_i as follows:

$$r_1 = \frac{1}{N_B}\sum_{t=1}^{N_B} q_t', \quad r_2 = \frac{1}{N_W}\sum_{t=N_B+1}^{N_B+N_W} q_t'. \tag{3}$$

Note here that $r_2 \geq r_1$. Finally, the value p_t of each pixel P_t in T_i is modulated to obtain a new pixel value p_t' by the following rule:

$$\textit{set } p_t' = r_1 \textit{ if } P_t \textit{ is black; or } r_2 \textit{ if } P_t \textit{ is white.} \tag{4}$$

The mean $\mu_{T_i'}$ of the pattern block T_i' so modulated will be equal to the mean μ_{B_i} of the target block B_i because we have:

$$\mu_{B_i} = \frac{1}{m}\sum_{t=1}^{m} q_t' = \frac{1}{N_B+N_W}\sum_{t=1}^{N_B+N_W} q_t', \tag{5}$$

and

$$\begin{aligned}
\mu_{T_i'} &= \frac{1}{m}\sum_{t=1}^{m} p_t' = \frac{1}{N_B+N_W}\sum_{t=1}^{N_B+N_W} p_t' \\
&= \frac{1}{N_B+N_W}\sum_{t=1}^{N_B} p_t' + \frac{1}{N_B+N_W}\sum_{t=N_B+1}^{N_B+N_W} p_t' \\
&= \frac{1}{N_B+N_W}\sum_{t=1}^{N_B} r_1 + \frac{1}{N_B+N_W}\sum_{t=N_B+1}^{N_B+N_W} r_2 \\
&= \frac{1}{N_B+N_W} N_B r_1 + \frac{1}{N_B+N_W} N_W r_2 \\
&= \frac{1}{N_B+N_W} N_B \Big(\frac{1}{N_B}\sum_{t=1}^{N_B} q_t'\Big) + \frac{1}{N_B+N_W} N_W \Big(\frac{1}{N_W}\sum_{t=N_B+1}^{N_B+N_W} q_t'\Big) \\
&= \frac{1}{N_B+N_W}\sum_{t=1}^{N_B+N_W} q_t' = \mu_{B_i}.
\end{aligned} \tag{6}$$

This means that the *overall* gray appearance of the modulated pattern image I_P' and that of the Y-component of I_T is roughly the same, as already mentioned. Accordingly, we replace the Y-component of I_T with I_P' to generate finally the desired signal-rich-art code image I_C which has the visual color appearance of I_T, as shown by the example seen in Fig. 6(f).

Later, when conducting message extraction, the message bit stream can be extracted from the Y-component of a captured version of I_C by classifying the pixels of each pattern block into two classes according to their Y values: black and white. However, if the two representative values r_1 and r_2 are too close, it will be difficult to "separate" them in the classification process. Therefore, an adjustment of the representative values r_1 and r_2 is conducted, resulting in r_1' and r_2', so that the absolute difference between r_1' and r_2' becomes not smaller than a pre-defined threshold $\delta \geq 0$. For example, Fig. 7 shows a pattern block resulting from modulations with different values of δ, from which one can see that the two colors in a modulated pattern block will be more easily distinguished when δ is larger.

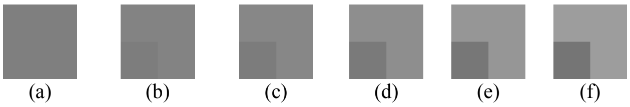


Fig. 7. Modulated pattern block resulting from uses of different threshold values of δ for the absolute difference between the two adjusted representative values r_1' and r_2'. (a) $\delta = 0$. (b) $\delta = 5$. (c) $\delta = 10$. (d) $\delta = 20$. (e) $\delta = 30$. (f) $\delta = 40$.

The detail of the proposed representative-value adjustment scheme is described in the following. Note that, after the adjustment, the absolute difference between r_1' and r_2' must be not smaller than the threshold δ, and that the mean of the modulated pattern block T_i'' based on r_1' and r_2' must be identical to that of the target block B_i. Thus, the values of r_1' and r_2' must satisfy the following two constraints:

$$|r_2' - r_1'| \geq \delta; \tag{7}$$

$$\mu_{T_i''} = \mu_{B_i}. \tag{8}$$

Two possible cases can be identified in the adjustment process: 1) the original absolute difference δ_o of r_1 and r_2 is already not smaller than δ, i.e., $\delta_o \geq \delta$; and 2) the reverse, i.e., $\delta_o < \delta$. In the first case, the values of r_1 and r_2 satisfy (7) and (8) automatically, so that they may be used as r_1' and r_2', respectively, directly, i.e., we have the rule:

$$\textit{if } \delta_o \geq \delta, \textit{ then set } r_1' = r_1 \textit{ and } r_2' = r_2. \tag{9}$$

For the second case with $\delta_o < \delta$, the absolute difference between the two representative values must be increased, after the adjustment, *at least* for the amount of $\delta - \delta_o$ for the resulting values of r_1' and r_2' to satisfy constraint (7). Specifically, let the adjustment value of r_1 be t so that $r_1' = r_1 - t$. Then, the adjustment value of r_2 should be at least $(\delta - \delta_o) - t$ so that $r_2' \geq r_2 + [(\delta - \delta_o) - t]$. Such value adjustments indeed can satisfy constraint (7) because with the fact that $r_2 - r_1 = \delta_o$, we have

$$|r_2' - r_1'| \geq |r_2 - r_1 + (\delta - \delta_o)| \geq \delta.$$

Also, for r_1' and r_2' to satisfy constraint (8), suppose that r_2' is adjusted for *exactly* the amount of $(\delta - \delta_o) - t$. The reason to make this assumption is to reduce the color distortion in the created signal-rich-art code image, as will be discussed later in Section 5. Then, the yet-unknown value t may be computed by:

$$\begin{aligned}
\mu_{T_i''} &= \frac{1}{m}\sum_{t=1}^{m} p_t'' = \frac{1}{N_B+N_W}\sum_{t=1}^{N_B} p_t'' + \frac{1}{N_B+N_W}\sum_{t=N_B+1}^{N_B+N_W} p_t'' \\
&= \frac{1}{N_B+N_W}\sum_{t=1}^{N_B} r_1' + \frac{1}{N_B+N_W}\sum_{t=N_B+1}^{N_B+N_W} r_2' \\
&= \frac{1}{N_B+N_W} N_B r_1' + \frac{1}{N_B+N_W} N_W r_2' \\
&= \frac{1}{N_B+N_W} N_B (r_1 - t) + \frac{1}{N_B+N_W} N_W (r_2 + [(\delta - \delta_o) - t]) \\
&= \frac{1}{N_B+N_W} N_B r_1 + \frac{1}{N_B+N_W} N_W r_2 - \frac{1}{N_B+N_W} N_B t
\end{aligned}$$

$$+\frac{1}{N_B+N_W}N_W(\delta-\delta_{\mathrm{o}})-\frac{1}{N_B+N_W}N_W t$$
$$=\mu_{B_i}-\frac{1}{N_B+N_W}N_B t+\frac{1}{N_B+N_W}N_W(\delta-\delta_{\mathrm{o}})-\frac{1}{N_B+N_W}N_W t$$
$$=\mu_{B_i} \qquad (10)$$

where the fact that the new value of p_t'' in T_i'' is set to be r_1' if the color of P_t in T_i is black; or to be r_2' if the color of P_t in T_i is white has been used in the derivations, and the last step is based on the use of (8). Accordingly, we can get:

$$-\frac{1}{N_B+N_W}N_B t+\frac{1}{N_B+N_W}N_W(\delta-\delta_{\mathrm{o}})-\frac{1}{N_B+N_W}N_W t=0\,, \quad (11)$$

from which the desired value t can be derived to be

$$t=[N_W/(N_B+N_W)](\delta-\delta_{\mathrm{o}})\,.$$

Therefore, the values of r_1' and r_2' can be computed by the rule:

if $\delta_{\mathrm{o}} < \delta$*, then set* $r_1' = r_1 - t$ *and* $r_2' = r_2 + [(\delta - \delta_{\mathrm{o}}) - t]$. (12)

C. *Algorithm for Signal-rich-art Code Image Creation*

Based on the above discussions, a detailed algorithm for signal-rich-art code image creation is described as follows.

Algorithm 1. ***Signal-rich-art code image creation.***

Input: a target image I_{T}, a message M, and a threshold value δ.

Output: a signal-rich-art code image I_{C}.

Steps:

Stage 1 — Transforming the message into a bit stream.

Step 1. Transform message M into a bit stream B.

Stage 2 — Generating the pattern image.

Step 2. Split B into n three-bit segments as $b_{11}b_{12}b_{13}b_{21}b_{22}b_{23}\ldots b_{n1}b_{n2}b_{n3}$.

Step 3. Expand every three bits $b_{i1}b_{i2}b_{i3}$ in B into four bits $b_{i1}'b_{i2}'b_{i3}'b_{i4}'$ according to (1) and generate the corresponding pattern block T_i according to the rules shown in Fig. 5.

Step 4. Align all the generated pattern blocks T_i in a raster-scan order to form a pattern image I_{P} of the size of target image I_{T}, with each side having N_T patterns; and if the result does not fill up I_{P}, repeat the filling until it becomes so.

Stage 3 — Modulating the pattern image.

Step 5. Divide the Y-component of target image I_{T} into *target blocks* $\{B_1, B_2, B_3, \ldots, B_N\}$ where $N = N_T \times N_T$.

Step 6. For each pattern block T_i in pattern image I_{P}, generate a modulated pattern block T_i'' as follows.

(A) Compute two representative values r_1 and r_2 of the corresponding target block B_i according to (3).

(B) Compute $\delta_{\mathrm{o}} = |r_2 - r_1|$, and use it and the input threshold δ to obtain two adjusted representative values r_1' and r_2' from r_1 and r_2 according to (9) and (12).

(C) For each pixel P_t in T_i, if P_t is black, set the value p_t'' of the corresponding pixel P_t'' in T_i'' as $p_t'' = r_1'$; else, set $p_t'' = r_2'$.

Step 7. Compose all the resulting T_i'' to get a modulated pattern image, denoted by I_{P}'.

Stage 4 — Injecting the pattern image into the target image.

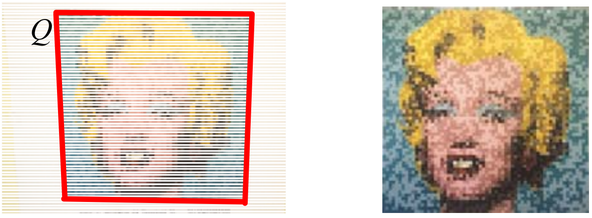
Step 8. Replace the Y-component of I_{T} with I_{P}' to generate the desired signal-rich-art code image I_{C} as the output.

IV. Message Extraction

The various techniques proposed for extracting the message embedded in the signal-rich-art code image are described first, with a combination of them described as an algorithm at last.

A. *Localization of Signal-rich-art Code Image and Inverse Perspective Transform*

Assume that the signal-rich-art code image I_{C} is printed and posted or displayed against a white background, and that the captured image I_{d} contains only the original image of I_{C} and the background. The first assumption here may be removed simply by adding a white surrounding zone to I_{C}. To extract the message from I_{d}, we must *localize* the region of I_{C} in I_{d}. For this, we apply the Hough transform and polygonal approximation to find the largest non-white quadrangle Q in I_{d} as shown by the example seen in Fig. 8(a). Also, image I_{d} will suffer from perspective distortion if the axis of the camera is not directed perpendicularly toward the plane of the signal-rich-art code image I_{C} [5] during image acquisition, as seen in Fig. 8(a) as well. As a remedy, an *inverse perspective transform* is performed on Q to correct the distortion. The result of conducting this on Fig. 8(a) is shown in Fig. 8(b). Finally, the Y-component of the resulting Q is taken as an intermediate result, which we call the *captured modulated pattern image* and denote it by I_{P}''.



(a) (b)

Fig. 8. Localization and correction of perspective distortion in captured signal-rich-art code image. (a) Localized signal-rich-art code image portion (enclosed by red rectangle). (b) Result of perspective distortion correction applied to red portion region in (a).

B. *Block Number Identification and Block Segmentation*

To identify the unit blocks in I_{P}'' in order to apply binarization and pattern recognition to them, an idea similar to the Hough transform [17] is adopted, which uses the statistics of the pixels' gradient values to *guess* the number N_{S} of unit blocks in the horizontal or vertical direction in I_{P}'' because those pixels on the splitting lines between the unit blocks usually have larger gradient values. In more detail, at first the gradient value g_{xy} of each pixel R_{xy} with value r_{xy} at coordinates (x, y) in I_{P}'' is computed by a Sobel operator [23]:

$$g_{xy}=\left|(r_{x+1,y-1}+2r_{x+1,y}+r_{x+1,y+1})-(r_{x-1,y-1}+2r_{x-1,y}+r_{x-1,y+1})\right|+$$
$$\left|(r_{x-1,y+1}+2r_{x,y+1}+r_{x+1,y+1})-(r_{x-1,y-1}+2r_{x,y-1}+r_{x+1,y-1})\right|\,. \quad (13)$$

Next, for each *possible* value n_j of N_{S}, the distance d_j between the splitting lines of every two possible adjacent unit blocks is computed as $d_j = L/n_j$ where L is the side length of the square-shaped I_P''. Then, the horizontal or vertical lines separated by the distance of d_j are taken as *candidate* splitting lines, where the positions of these candidate splitting lines described by image coordinates are computed by:

$$x = u\times d_j \textit{ and } y = v\times d_j, \tag{14}$$

where $u = 1 \sim \lfloor L/d_j\rfloor$ and $v = 1 \sim \lfloor L/d_j\rfloor$, respectively. Also, the *average gradient value* AG_{n_j} of the pixels on each candidate spitting line is computed as:

$$AG_{n_j}=\frac{1}{\lfloor L/d_j\rfloor\times L}\sum_{u=1}^{\lfloor L/d_j\rfloor}\sum_{y=1}^{L} g_{(u\times d_j)y}+\frac{1}{\lfloor L/d_j\rfloor\times L}\sum_{v=1}^{\lfloor L/d_j\rfloor}\sum_{x=1}^{L} g_{x(v\times d_j)}. \tag{15}$$

Note that in addition to the actual value N_S which will yield a large average gradient value, the values n_j which are divisors of N_S will yield large average gradient values as well. For this, the value n_j yielding the *largest* average gradient value AG_{n_j}, denoted as AG_o, is selected first and those n_j yielding average gradient values AG_{n_j} *close* to AG_o are selected as well. Then, the largest n_j from these selected values of n_j is taken as the desired number N_S of blocks of I_P'' in the horizontal or vertical direction, and division of I_P'' into unit blocks is conducted accordingly.

For example, Fig. 9(a) shows a captured modulated pattern image I_P'', Fig. 9(b) is the image of the computed gradient values, and Fig. 9(c) illustrates the average gradient values for different values of N_S, where the n_j yielding the largest average gradient value is seen to be 16 (indicated by the black arrow). Also, the n_j's, which yield the average gradient values close to the largest average gradient value 16, are seen to be 32 and 64 (indicated by the green and red arrows, respectively), and the n_j yielding the largest gradient value among the three selected ones is 64. Therefore, the desired N_S is taken to be 64, and the corresponding image division result is shown in Fig. 9(d).

C. Binarization and Recognition of Pattern Blocks

After the captured modulated pattern image I_P'' are segmented into unit blocks, we try to recover the pattern blocks in pattern image I_P by grouping every four mutually-connected unit blocks as a pattern block since the size of a pattern block is 2×2. The number of pattern blocks in the horizontal or vertical direction in I_P'' is so just $N_T = N_S/2$. Subsequently, the moment-preserving thresholding technique [18] is applied to each pattern block T_i''' to binarize it automatically. And the four *unit blocks* in each resulting pattern block T_i''' are denoted as F_{i1}', F_{i2}', F_{i3}', and F_{i4}', respectively.

Next, how to classify each T_i''' as one of the eight possible *code patterns*, which we denote as BP_k with $k = 1 \sim 8$, as shown in Fig. 5 is the issue now. This is an eight-class pattern classification problem. To solve it, we use a *minimum absolute distance classifier*. Specifically, each possible code pattern BP_k has four unit blocks, say denoted as F_{k1}'' through F_{k4}'', and the color of each F_{kj}'' is either black or white. Hence, we may utilize the *feature of blackness* to describe F_{kj}''; that is, if the color of the unit block F_{kj}'' is black, then we take the blackness feature bf_{kj} of F_{kj}'' to be "1"; else, to be "0." Next, we compute the *real* blackness feature bf_{ij} of each unit block F_{ij}' in T_i''' by:

$$bf_{ij} = NB_{ij}/(NB_{ij} + NW_{ij}), \tag{16}$$

where NB_{ij} and NW_{ij} are the numbers of black and white pixels in unit block F_{ij}', respectively. Then, the *absolute distance* AD_{ik} of the blackness feature between T_i''' and BP_k can be computed as:

$$AD_{ik} = \frac{1}{4}\sum_{j=1}^{4}\left|bf_{ij} - bf_{kj}\right|. \tag{17}$$

Subsequently, the code pattern BP_m with the minimum absolute distance AD_{im} is selected as the result of classifying the pattern block T_i'''. For example, let the blackness features of unit blocks F_{i1}' through F_{i4}' of a pattern block T_i''' be $bf_{i1} = 0.9$, $bf_{i2} = 0.13$, $bf_{i3} = 0.22$, $bf_{i4} = 0.12$, respectively. The absolute distances AD_{ik} of T_i''' to all the eight possible code patterns BP_k with $k = 1 \sim 8$ are shown in Table I. And the code pattern with the minimum absolute distance is BP_4 with $AD_{i4} = 0.1425$. So, the corresponding four bits b_{i1}'' through b_{i4}'' of the pattern block T_i''' are "0111." Finally, we take the three bits b_{i1}'', b_{i2}'', and b_{i3}'', namely, "011," as the recovered version of the original three message bits b_{i1}, b_{i2}, and b_{i3} according to the bit expansion scheme.

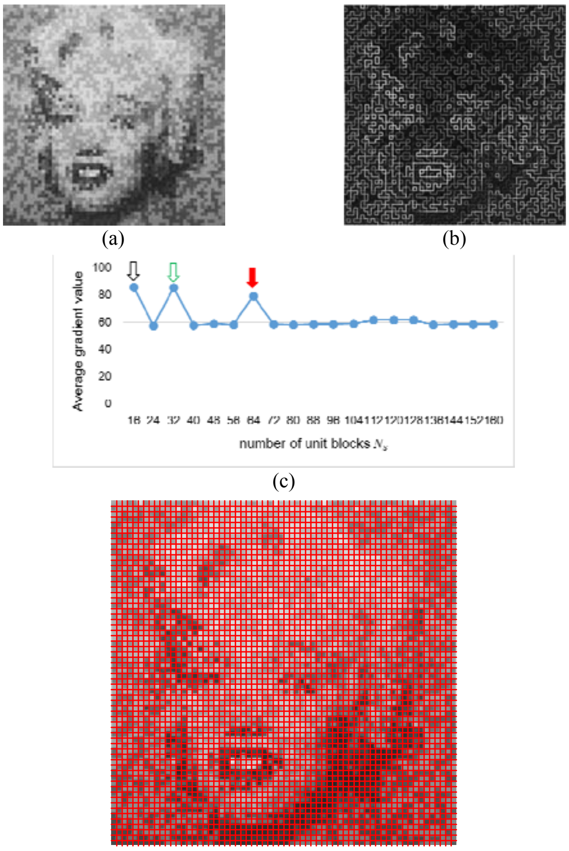
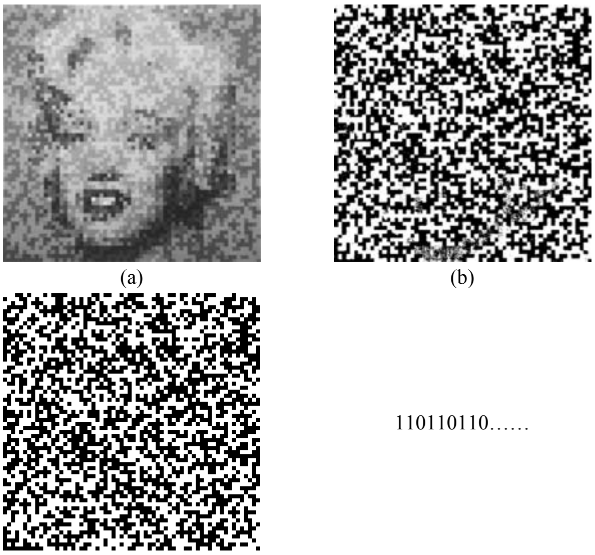


Fig. 9. Block number identification. (a) Captured modulated pattern image I_P''. (b) Gradient values of (a). (c) Average gradient values of pixels on candidate spitting lines for different N_S. (d) Image division result according to determined number of unit blocks, $N_S = 64$.

An example of results yielded by the above message extraction process is shown in Fig. 10, where a captured modulated pattern image I_P'' is shown in Fig. 10(a), which, after being binarized, results in Fig. 10(b); the result of code-pattern classification is shown in Fig. 10(c); and the final extracted bit stream are shown in Fig. 10(d). These results show that the proposed code-pattern classification scheme corresponding to the minimum absolute distance criterion works correctly for the purpose of embedded message extraction. More experimental results will be presented later to prove this statement.

TABLE I
AN EXAMPLE OF CODE PATTERN RECOGNITION

binary code pattern	Corresponding 4 bits	Absolute distance AD_{ik}
	0001	$AD_{i1} = (0.1+0.87+0.78+0.12)/4 = 0.4675$
	0010	$AD_{i2} = (0.1+0.87+0.22+0.88)/4 = 0.5175$
	0100	$AD_{i3} = (0.1+0.13+0.78+0.88)/4 = 0.4725$
	0111	$\mathbf{AD_{i4} = (0.1+0.13+0.22+0.12)/4 = 0.1425}$
	1000	$AD_{i5} = (0.9+0.87+0.78+0.88)/4 = 0.8575$
	1011	$AD_{i6} = (0.9+0.87+0.22+0.12)/4 = 0.5275$
	1101	$AD_{i7} = (0.9+0.13+0.78+0.12)/4 = 0.4825$
	1110	$AD_{i8} = (0.9+0.13+0.22+0.88)/4 = 0.5325$



(a) (b) (c) (d)

Fig. 10. Binarization and code-pattern recognition. (a) Captured modulated pattern image. (b) Binarization result of (a). (c) Result of code-pattern recognition of (b). (d) Extracted message.

D. Message Extraction Algorithm

A detailed message extraction algorithm is as follows.

Algorithm 2. ***Message extraction.***

Input: a captured version I_d of a signal-rich-art code image.

Output: a message M' extracted from I_d, which is supposed to be identical to the original message M' embedded in I_d.

Steps:

Stage 1 — Localizing the signal-rich-art code image.

Step 1. Find the largest non-white quadrangle Q in I_d by the Hough transform and polygonal approximation.

Stage 2 — Correcting geometric distortion.

Step 2. Perform an inverse perspective transform on Q to correct the perspective distortion and take the Y-component of Q as the captured modulated pattern image I_P''.

Stage 3 — Identifying pattern blocks in the code image.

Step 3. Compute the gradient value g_{xy} of each pixel R_{xy} in I_P'' according to (13).

Step 4. For each possible value n_j of N_S, compute the average gradient values AG_{n_j} of the pixels on each candidate spitting line according to (15).

Step 5. Select the value n_j yielding the largest AG_{n_j}, denoted as AG_o, and those n_j's yielding AG_{n_j} close to AG_o; pick the largest n_j from all the selected n_j's for use as the desired number N_S of blocks of I_P'' in the horizontal or vertical direction; and divide I_P'' accordingly into unit blocks.

Stage 4 — Binarizing the pattern blocks to extract the message.

Step 6. Group every four mutually-connected unit blocks and denote them as F_{i1}' through F_{i4}' to form a pattern block T_i''' in I_P''.

Step 7. Extract three bits b_{i1}'', b_{i2}'', and b_{i3}'' from each pattern block T_i''' by the following steps.

(A) For each unit block F_{ij}' in T_i''', compute its blackness feature bf_{ij} according to (16).

(B) Computing the absolute distance AD_{ik} of T_i''' to each of the eight possible code patterns BP_k shown in Fig. 5 according to (17).

(C) Select the code pattern BP_m with the minimum absolute distance and take the corresponding four bits of BP_m as the recognized four bits b_{i1}'' through b_{i4}'' of T_i'''.

(D) Take the three bits b_{i1}'', b_{i2}'', and b_{i3}'' as the recovered version of the original three message bits b_{i1}, b_{i2}, b_{i3}, respectively.

Step 8. Concatenate the extracted bits into a bit stream B and transform reversely B to get the embedded message M'.

V. EXPERIMENTAL RESULTS

The proposed method was implemented on a 3.0GHz PC with a Core i7 CPU and 8G RAM using the language Microsoft C#.NET, and generated signal-rich-art code images were captured with an iPhone 4S and analyzed to extract the embedded messages in a series of experiments. The resolution of the captured image is 800M pixels and the modulated pattern image I_P'' is down-sampled to be 640×640 in size (e.g., one bit of information in Fig. 10(b) corresponds to 10×10 pixels) in order to reduce the processing time. Corresponding statistics were plotted as well to show the accuracy of the extracted messages (*the number of correct bits/the total number of bits*) using different parameters including: (1) the threshold δ for the minimum difference between the representative values r_1' and r_2'; and (2) the number of unit blocks N_S used in the horizontal or vertical direction of the created pattern image. Figs. 11(a), 11(c), and 11(e) show three test target images used in the experiments. The corresponding signal-rich-art code images generated with parameters $N_S = 128$ and $\delta = 40$ are shown in Figs. 11(b), 11(d), and 11(f), respectively. These images were all printed to be of the same size of 127×127 mm.

One of the parameters that influence the accuracy of the extracted message is the threshold value δ for the minimum difference between the two representative values r_1' and r_2'. If δ is too small, r_1' and r_2' will be too close so that the extracted message might be wrong. Fig. 13(a) illustrates the accuracy rates of message extraction with $\delta = 0$, 20, 40, and 60, which shows that the larger the value of δ, the higher the accuracy rate of the extracted message; when $\delta > 40$, an accuracy of 99.8% is reached; and when $\delta > 60$, an accuracy of 100% is reached. Fig. 13(b) shows that the larger the value of δ, the smaller the PSNR (the *peak signal-to-noise ratio*) of the resulting signal-rich-art code image with respect to the target image. So there is a tradeoff between achieving higher message extraction accuracy and obtaining better visual quality in the generated code image. Fig. 12 shows some code images created with different threshold values of δ, where the target image is Fig. 11(e) and $N_S = 64$. As can be seen, Fig. 12(a) has the best visual appearance when compared with the others, but has the lowest message extraction accuracy for only 85.60%, because the two representative values are too close (so that the colors of most regions in Fig. 12(a) look like the same).

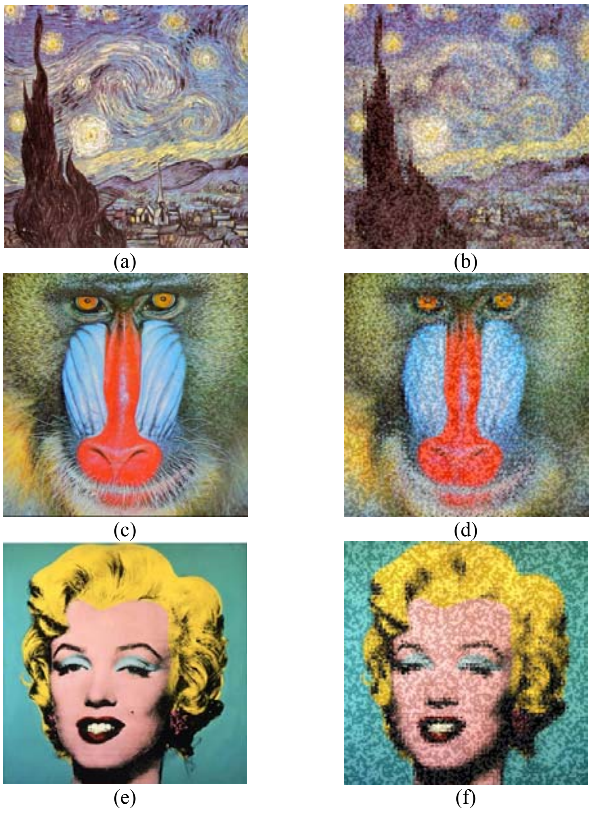


Fig. 11. Created signal-rich-art code images. (a), (c), and (e) Test target images. (b), (d) and (f) Resulting signal-rich-art code images with $N_S = 128$ and $\delta = 40$.

Another parameter that influences the message extraction accuracy is the number N_S of unit blocks in the horizontal or vertical direction in the created pattern or code image. The larger the value of N_S, the larger the message embedding capacity of the created code image, yet the smaller the size of the unit block and so the lower the message extraction accuracy. This can be seen from Fig. 13(c), where when $N_S = 16$, the accuracy of 100% is reached; when $N_S = 64$, the accuracy of 99.76% is reached; and when $N_S = 128$, the lower accuracy of 96.31% is yielded. Fig. 14 shows some signal-rich-art code images generated with different values of N_S with Fig. 11(a) as the target image and the threshold $\delta = 40$. As can be seen, when N_S is larger, the visual appearance of the created image is better with a larger PSNR, but the message extraction accuracy is lower. Specifically, the accuracy of Fig. 14(d) is 99.11%, instead of 100% which is reached by the other three cases.

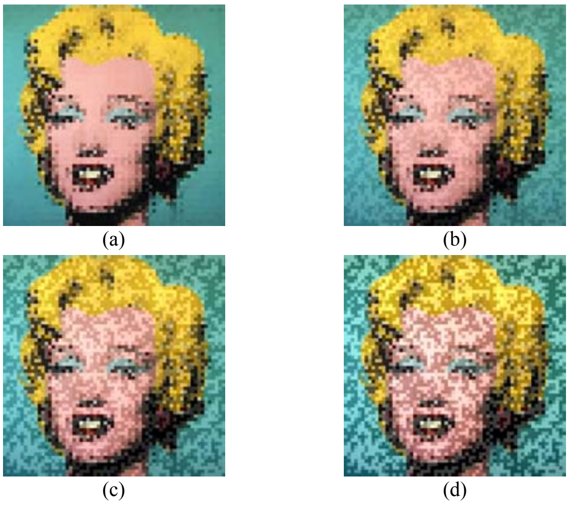
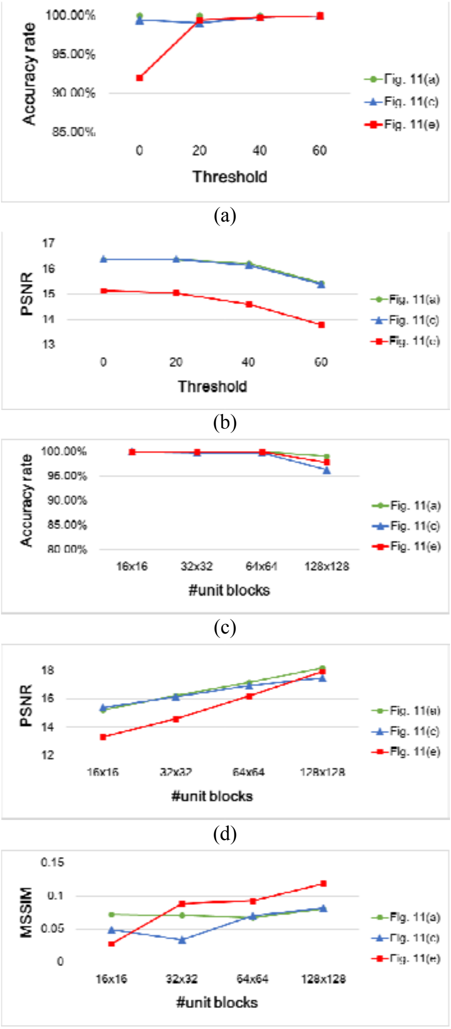


Fig. 12. Created signal-rich-art code images with different threshold values of δ, where $N_S = 64$. (a) Resulting signal-rich-art code image with PSNR = 17.38 and accuracy rate = 85.60%, where $\delta = 0$. (b) Resulting code image with PSNR = 17.12 and accuracy rate = 98.97%, where $\delta = 20$. (c) Resulting code images with PSNR = 16.19 and accuracy rate = 100%, where $\delta = 40$. (b) Resulting images with PSNR = 14.83 and accuracy rate = 100%, where $\delta = 60$.

Moreover, to measure the *blockiness effect* seen in the generated signal-rich-art code images, we adopt the metric of mean structural similarity (MSSIM) to compare the similarity between the created code image and the target image [25]. Fig. 13(e) shows such MSSIM values versus different numbers N_S of unit blocks, where the window size for computing the MSSIM is set to be the same as the size of a pattern block. We can see from Fig. 13(e) that the MSSIM value is larger when N_S is larger, implying that the visual appearance of the created image is better (i.e., blockiness effect is smaller) when N_S is larger.

Table II shows a comparison of the results of the proposed method and those of Lee and Tsai [15] with the target images as shown in Fig. 11 against different numbers of unit blocks in the horizontal or vertical direction in the created pattern or message images. As can be seen from the table, the proposed method yields higher message extraction accuracy than Lee and Tsai [15], e.g., when $N_S = 32$, the message extraction accuracy yielded by the proposed method reaches 99.80% while that yielded by Lee and Tsai [15] is only 88.28%. Moreover, when $N_S = 64$, the message extraction accuracy of 99.76% yielded by the proposed method is *much higher* than that yielded by Lee and Tsai [15], which is only 34.70%. However, comparing the two methods, the amount of information contained in a code image created by the proposed method is less, as shown in Table II as well. Specifically, the amount of information contained in an image created by the proposed method is equal to just a half of that contained in the corresponding image created by Lee and Tsai [15] when the values of N_S used in the two methods are identical (because a unit block represents one bit in the proposed method and 1/4 character, or two bits, in Lee and Tsai [15], respectively). Also, Fig. 15 shows a comparison of accuracy rates versus different amounts of information contained in an image created by the proposed method and Lee and Tsai [15], where, as can be seen, the accuracy rate of the proposed method is still good when the amount of information is 4096 while the accuracy rate of Lee and Tsai [15] becomes very bad when the amount of information is 8192.



(a)

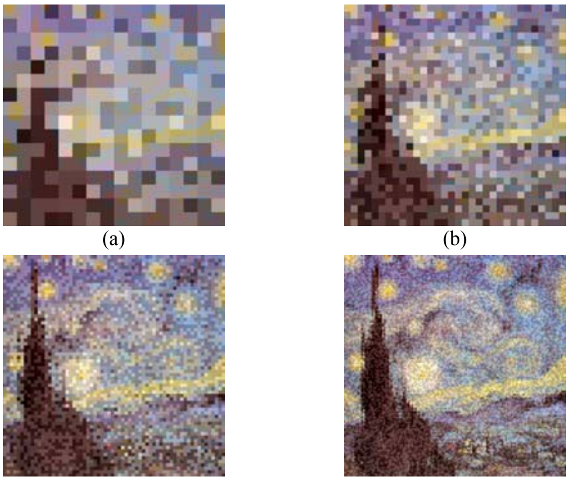
(b)

(c)

(d)

(e)

Fig. 13. Plots of trends of results using various parameters. (a) Accuracy rates of extracted messages with different threshold values δ, with #unit blocks N_S = 32. (b) RMSE values of created signal-rich-art code images with respect to target images for different threshold values of δ, with #unit blocks N_S = 32. (c) Accuracy rates of extracted messages with different #unit blocks N_S with threshold δ = 40. (d) RMSE values of created signal-rich-art code images with respect to target images with different #unit blocks N_S and threshold δ = 40. (e) MSSIM values of created signal-rich-art code images with respect to target images with different N_S of unit blocks and threshold δ = 40.



(a)

(b)

(c)

(d)

Fig. 14. Created signal-rich-art code images with different #unit blocks N_S, where threshold value δ = 40. (a) Resulting signal-rich-art code image with RMSE = 47.66 and accuracy rate = 100%, where N_S = 16. (b) Resulting signal-rich-art code image with RMSE = 44.63 and accuracy rate = 100%, where N_S = 32. (c) Resulting signal-rich-art code image with RMSE = 42.05 and accuracy rate = 100.00%, where N_S = 64. (d) Resulting signal-rich-art code image with RMSE = 39.43 and accuracy rate = 99.11%, where N_S = 128.

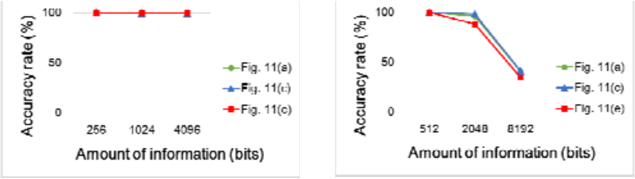
Also, Fig. 16 shows the resulting binarized captured signal-rich-art images of the proposed method and Lee and Tsai [15], where the value of N_S is 32 in Figs. 16(a) and 16(b) and 64 in Figs. 16(c) and 16(d). As can be seen from Fig. 16(a), with N_S = 32 the characters in the binarized captured signal-rich-art character image created by Lee and Tsai [15] are still clear enough so that the message extraction accuracy yielded with Fig. 16(a) as the input is still high, reaching 98.61%. However, with N_S = 64, as seen from Fig. 16(c), the characters in the binarized captured signal-rich-art character image created by Lee and Tsai [15] become undistinguishable so that the message extraction accuracy yielded with Fig. 16(c) as the input becomes worse, only 41.25%. Furthermore, as seen from Figs. 16(b) and 16(d), the pattern blocks in the binarized captured signal-rich-art code images created by the proposed method are both clear enough so that the message extraction accuracy rates yielded by them are both still high, reaching 99.80% and 99.76%, respectively.

In addition, we compare the times consumed by the code-pattern recognition steps in the proposed method and Lee and Tsai [15]. As can be seen from Table II, the recognition time used by the proposed method is much less than that used by Lee and Tsai [15]. This is owing to the time-consuming OCR operation conducted by Lee and Tsai [15] on every character image, which computes the similarity of the character image with each possible character image in the database and selects the most similar one as the recognition result. In contrast, the proposed method only needs to recognize each pattern block as coming from one of eight possible code-pattern classes by computing the absolute distances of the pattern block to the eight classes and selecting the one with the minimum absolute difference.

As a summary, the proposed method has the following merits with respect to Lee and Tsai [15]: (1) the yielded signal-rich-art code image has a better visual appearance since a larger number N_S of unit blocks can be utilized in the proposed method; (2) the message extraction accuracy is higher since much less details are contained in a unit block of the proposed method; (3) the message extraction speed is higher since classification of only eight classes need be conducted to extract the corresponding four bits of each binarized pattern block.

TABLE II
COMPARISON OF RESULTS OF PROPOSED METHOD AND LEE AND TSAI [15] WITH δ = 40.

Target image	N_S	Method	Accuracy rate (%)	Recognition time (ms)	Amount of information (bits/image)
Fig. 11(a)	16	Proposed method	100	**60**	256
		Lee and Tsai [15]	100	1186	512
	32	Proposed method	**100**	**62**	1024
		Lee and Tsai [15]	96.53	1812	2048
	64	Proposed method	**100**	**93**	4096
		Lee and Tsai [15]	40.86	3045	8192
Fig. 11(c)	16	Proposed method	100	**68**	256
		Lee and Tsai [15]	100	1590	512
	32	Proposed method	**99.80**	**63**	1024
		Lee and Tsai [15]	98.61	1696	2048
	64	Proposed method	**99.76**	**83**	4096
		Lee and Tsai [15]	41.25	2263	8192
Fig. 11(e)	16	Proposed method	100	**54**	256
		Lee and Tsai [15]	100	1230	512
	32	Proposed method	**99.80**	**55**	1024
		Lee and Tsai [15]	88.28	1697	2048
	64	Proposed method	**100**	**82**	4096
		Lee and Tsai [15]	34.70	2315	8192



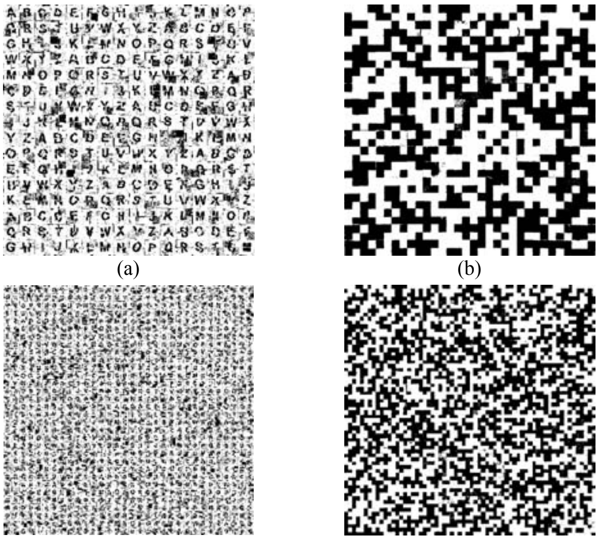
(a) (b)

Fig. 15. Comparison of accuracy rates versus different amounts of information contained in an image created the proposed method and Lee and Tsai [15]. (a) Accuracy rate versus different amount of information contained in an image of proposed method. (b) Accuracy rate versus different amount of information contained in an image of Lee and Tsai [15].

Furthermore, it is noted that distortions may be incurred during message extraction due to perspective deformation, noise addition, blurring, uneven lighting, etc. We have solved the problems of perspective deformation and uneven lighting in the proposed method. Also, the issues of noise addition (such as uniform-distribution or burst errors) and blurring have also been dealt with by the proposed method by code pattern recognition. However, how to overcome the interference during message extraction is still an open research area since many types of noise exist in our environment. One possible way may be directed to applying error-correction techniques to the result of code-pattern classification in order to increase the resulting message extraction rate, such as using Reed-Solomon codes [26].

VI. CONCLUSIONS

A new type of signal-rich-art image for applications of data transfer, called *signal-rich-art code image*, has been proposed, which is created from a target image for use as a carrier of a given message. The artistic favor of the target image is kept in the created image, achieving the signal-rich-art effect. Skillful techniques of code pattern design, unit block segmentation, pattern block classification, etc. have been proposed for message data embedding and extraction. Comparing with other signal-rich-art techniques like the use of barcodes and data hiding, data transfer using the proposed signal-rich-art code image has several merits: (1) the image has the visual appearance of any pre-selected target image (this is not the case for the case of using barcodes [19]-[22]); (2) the proposed method can endure more distortions in acquired versions of the code image like perspective transformation, noise, screen blurring, etc. (this is not the case for data hiding [7]-[14]); (3) the message can be extracted from an image captured by a mobile device (this is not the case for data hiding [7]-[14]). Also, the proposed method has following additional merits when compared with Lee and Tsai [15], in which another type of signal-rich-art image, called signal-rich-art character image was proposed: (1) the yielded signal-rich-art code image has a better visual appearance; (2) the message data extraction accuracy is higher; (3) the data extraction speed is higher. Experimental results show the feasibility of the proposed method. Further studies may be directed to designing more types of signal-rich-art images or extending the idea to deal with videos for different applications.



(a) (b) (c) (d)

Fig. 16. Binarized captured signal-rich-art images created by Lee and Tsai [15] and proposed method and respective message extraction accuracy rates, where the target image of these resulting images is Fig. 11(c). (a) Binarized image by Lee and Tsai [15] with N_S = 32 and accuracy rate = 98.61%. (b) Binarized image by proposed method with N_S = 32 and accuracy rate = 99.80%. (c) Binarized image by Lee and Tsai [15] with N_S = 64 and accuracy rate = 41.25%. (d) Binarized image by proposed method with N_S = 64 and accuracy rate = 99.76%.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for many useful comments as well as suggestions for using the odd parity in bit expansion and comparing the amounts of information contained in created images yielded by the proposed method

and by Lee and Tsai [15].

REFERENCES

[1] B. Davis, “Signal rich art: enabling the vision of ubiquitous computing,” *Proc. SPIE* 7880: *Media Watermarking, Security, and Forensics III*, N. D. Memon, J. Dittmann, A. M. Alattar, and E. J. Delp III, Eds., vol. 788002, Feb. 2011.

[2] S. Poslad, *Ubiquitous Computing: Smart Devices, Environments and Interactions*, John Wiley & Sons, Chichester, UK, 2009.

[3] E. Ouaviani, A. Pavan, M. Bottazzi, E. Brunclli, F. Caselli, and M. Guerrero, “A common image processing framework for 2D barcode reading,” in *7th Int. Conf. on Image Process. and Its Appl.*, vol. 2, no. 465, pp. 652–655, Jul. 1999.

[4] C. Zhang, J. Wang, S. Han, M. Yi and Z. Zhang, “Automatic real-time barcode localization in complex scenes,” in *IEEE Int. Conf. on Image Processing*, pp. 497-500, 2006.

[5] H. Yang, A. C. Kot, and X. Jiang, “Accurate localization of four extreme corners for barcode images captured by mobile phones,” *Proc. IEEE Int. Conf. on Image Processing*, pp. 3897-3900, 2010.

[6] H. Yang, A. C. Kot, and X. Jiang, “Binarization of low-quality barcode images captured by mobile phones using local window of adaptive location and size,” *IEEE Trans. Image Processing*, vol. 21, no. 1, pp. 418-425, 2012.

[7] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, “Reversible Data Hiding,” *IEEE Trans. Circuits Syst. & Video Technol.*, vol. 16, no. 3, pp. 354-362, March 2006.

[8] P. Y. Lin, J. S. Lee and C. C. Chang, “Protecting the content integrity of digital imagery with fidelity preservation,” *ACM Trans. Multimedia Computing Communications and Applications*, vol. 7, no. 3, 2011.

[9] R. L. Lagendijk, G. C. Langelaar, and I. Setyawan, “Watermarking digital image and video data,” *IEEE Signal Proc. Mag.*, vol. 17, pp. 20-46, Sept. 2000.

[10] G. Doërr and J.-L. Dugelay, “A guide tour of video watermarking,” *Signal Processing: Image Commun.*, vol. 18, no. 4, pp.263-282, 2003.

[11] W. N. Lie and L. C. Chang, “Robust and high-quality time-domain audio watermarking based on low-frequency amplitude modification,” *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 46-59, 2006.

[12] O. Bulan, G. Sharma, and V. Monga, “Orientation modulation for data hiding in clustered-dot halftone prints,” *IEEE Trans. Image Processing*, vol. 19, no. 8, pp. 2070-2084, 2010.

[13] O. Bulan, and G. Sharma, “High capacity color barcodes: per channel data encoding via orientation modulation in elliptical dot arrays,” *IEEE Trans. Image Processing*, vol. 20, no. 5, pp. 1337-1350, 2011.

[14] N. Damera-Venkata, J. Yen, V. Monga, and B. L. Evans, “Hardcopy image barcodes via block-error diffusion,” *IEEE Trans. Image Processing*, vol. 14, no. 12, pp. 1977-1989, 2005.

[15] Y. L. Lee and W. H. Tsai, “Signal rich art image — a new tool for automatic identification and data capture applications using mobile phones,” *Proc. IEEE Intl. Conf. Acoustics Speech and Sig. Proc.*, pp. 1942-1946, 2013.

[16] C. Lin, “Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network,” *Pattern Recognition Letters*, vol. 28, pp.2190-2200, 2007.

[17] J. Illingworth and J. Kittler, “A survey of the Hough transform,” *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87-116, 1998.

[18] W. H. Tsai, “Moment-preserving thresholding: a new approach,” *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377-393, 1985.

[19] *BS ISO/IEC 16388: Information Technology-Automatic Identification and Data Capture Techniques-Code39 Bar Code Symbology Specification*, BS ISO/IEC 16388, 2007.

[20] *BS ISO/IEC 15438: Information Technology-Automatic Identification and Data Capture Techniques-PDF417 Barcode Symbology Specification*, BS ISO/IEC 15438, 2006.

[21] *BS ISO/IEC 18004: Information Technology-Automatic Identification and Data Capture Techniques-QR Code 2005 Bar Code Symbology Specification*, BS ISO/IEC 18004, 2006.

[22] *BS ISO/IEC 16022: Information Technology-Automatic Identification and Data Capture Techniques-Data Matrix Bar Code Symbology Specification*, BS ISO/IEC 16022, 2006.

[23] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, pp. 572-580, 2002.

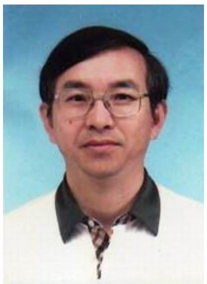
[24] A. Furness, “Machine-readable data carriers - a brief introduction to automatic identification and data capture,” *Assembly Automation*, vol. 20, pp. 28-34, 2000.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Trans. Image Processing, vol. 13, no. 4, pp. 600-612, 2004.

[26] V. Guruswami and M. Sudan, “Improved decoding of Reed-Solomon and algebraic-geometric codes,” *IEEE Trans. Inform. Theory*, vol. 45, pp.1757-1767, 1999.

Ya-Lin Lee received the B. S. degree in 2009 and the Ph. D. degree in 2013, both in computer science and information engineering from the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan.

She has been a research assistant at the Computer Vision Laboratory in the Department of Computer Science at National Chiao Tung University from August 2009 to October 2013. Currently, she is a Senior Engineer with MediaTek Inc., Hsinchu, Taiwan. Her current research interests include information hiding, image processing, pattern recognition, and artificial intelligence.



Wen-Hsiang Tsai received the B.S. degree in EE from National Taiwan University, Taiwan, in 1973, the M.S. degree in EE from Brown University, USA in 1977, and the Ph.D. degree in EE from Purdue University, USA in 1979. Since 1979, he has been with National Chiao Tung University (NCTU), Taiwan, where he is now a Chair Professor of Computer Science. At NCTU, he has served as the Head of the Dept. of Computer Science, the Dean of General Affairs, the Dean of Academic Affairs, and a Vice President. From 1999 to 2000, he was the Chair of the Chinese Image Processing and Pattern Recognition Society of Taiwan, and from 2004 to 2008, the Chair of the Computer Society of the IEEE Taipei Section in Taiwan. From 2004 to 2007, he was the President of Asia University, Taiwan.

Dr. Tsai has been an Editor or the Editor-in-Chief of several international journals, including *Pattern Recognition*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *Journal of Information Science and Engineering*. He has published 161 journal papers and 248 conference papers and received many awards, including the Annual Paper Award from the Pattern Recognition Society of the USA; the Academic Award of the Ministry of Education, Taiwan; the Outstanding Research Award of the National Science Council, Taiwan; the ISI Citation Classic Award from Thomson Scientific, and more than 40 other academic paper awards from various academic societies. His current research interests include computer vision, information security, video surveillance, and autonomous vehicle applications. He is a Life Member of the Chinese Pattern Recognition and Image Processing Society, Taiwan and a Senior Member of the IEEE.