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 advertisement 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.](image_url)

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,
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.

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 $I_Y$;
Step 1.4 — replace the Y-channel of target image $I_T$ with $I_Y$ 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 $I_C'$;
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.](image-url)
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
\[
B = b_{11}b_{21}b_{31}b_{32}b_{22}b_{23}b_{32}b_{33} ... b_{3n-2}b_{3n},
\]
and for every three consecutive bits \( b_{ij}b_{i}b_{j} \) in \( B \), we append an odd parity bit \( b_{ij}' \) to get four bits \( b_{ij}b_{i}b_{j}b_{ij}' \) according to the following rule:
\[
\text{set } b_{ij}' = b_{ij} \lor b_{i} \lor b_{j} \text{ and } b_{ij}' = b_{ij} \text{ for } j = 1, 2, 3,
\]
where \( \lor \) and \( \lnot \) denote bitwise “OR” and “complement” operations, respectively. The resulting four bits \( b_{ij}'b_{i}b_{j}b_{ij}' \) are not all identical, as can be verified by ORing them to get the following result:
\[
b_{ij}' \lor b_{i} \lor b_{j} \lor b_{ij}' = (b_{ij} \lor b_{i} \lor b_{j}) \lor b_{ij}' = b_{ij} \lor b_{i} \lor b_{j} = 1,
\]
which means that at least one “1” must exist in \( b_{ij}'b_{i}b_{j}b_{ij}' \), and if all the four bits are “1s,” then all the three bits \( b_{ij}, b_{i}, b_{j} \), and \( b_{ij}' \) must be 1’s, leading to the result \( b_{ij}' = b_{ij} \lor b_{i} \lor b_{j} = 0 \), which is a contradiction. Moreover, the total possible number of distinct expanded four bits \( b_{ij}'b_{i}b_{j}b_{ij}' \) for different combinations of \( b_{ij}, b_{i}, b_{j} \) 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 \( T_{i} = F_{i}F_{i}F_{i}F_{i} \) with four unit blocks \( F_{i} \) through \( F_{i} \) to represent the non-all-identical bits \( b_{ij}'b_{i}b_{j}b_{ij}' \) of each code pattern \( C_{i} \), where the color of unit block \( F_{i} \) 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.

<table>
<thead>
<tr>
<th>Message bits</th>
<th>Code pattern</th>
<th>Pattern block ( T_{i} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0000</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>001</td>
<td>0010</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>010</td>
<td>0100</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>011</td>
<td>0111</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>101</td>
<td>1011</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>110</td>
<td>1101</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
<tr>
<td>111</td>
<td>1110</td>
<td><img src="image" alt="Pattern block" /></td>
</tr>
</tbody>
</table>

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 \( B = “1010110100011110101111001...” \) 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.

![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.](image)
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 \{ $q_{1'}, q_{2'}, \ldots, q_{n'}$\}. Then, two representative Y values $r_1$ and $r_2$ are computed for $B_i$ as follows:

$$
r_1 = \frac{1}{N_b + N_w} \sum_{i=1}^{N_b} q_{1'}, \quad r_2 = \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} q_{1'}.
$$

(3)

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

$$
set p'_i = r_1 \text{ if } P_i \text{ is black; or } r_2 \text{ if } P_i \text{ is white.}
$$

(4)

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

$$
\mu_{bi} = \frac{1}{m} \sum_{i=1}^{m} q_i = \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} q_i,
$$

(5)

and

$$
\mu_i = \frac{1}{m} \sum_{i=1}^{m} p'_i = \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} p'_i
$$

$$
= \frac{1}{N_b + N_w} \sum_{i=1}^{N_b} q'_i + \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} p'_i
$$

$$
= \frac{1}{N_b + N_w} \sum_{i=1}^{N_b} q'_i + \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} q_i
$$

$$
= \frac{1}{N_b + N_w} \sum_{i=1}^{N_b+N_w} q_i = \mu_{bi}.
$$

(6)

This means that the overall gray appearance of the modulated pattern image $I'_p$ and that of the Y-component of $I_f$ is roughly the same, as already mentioned. Accordingly, we replace the two classes according to their Y values: black and white. This means that the Y-component of the same, as already mentioned. Accordingly, we replace the two classes according to their Y values: black and white.

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 $\delta$, from which one can see that the two colors in a modulated pattern block will be more easily distinguished when $\delta$ is larger.

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 $\delta$, 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:

$$
|r'_2 - r'_1| \geq \delta,
$$

(7)

$$
\mu_{i'} = \mu_{bi}.
$$

(8)

Two possible cases can be identified in the adjustment process: 1) the original absolute difference $\delta_i$ of $r_1$ and $r_2$ is already not smaller than $\delta$, i.e., $\delta_i \geq \delta$, and 2) the reverse, i.e., $\delta_i < \delta$. 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:

$$
\text{if } \delta_i \geq \delta \text{, then set } r'_1 = r_1 \text{ and } r'_2 = r_2.
$$

(9)

For the second case with $\delta_i < \delta$, the absolute difference between the two representative values must be increased, after the adjustment, at least for the amount of $\delta - \delta_i$ for the resulting values of $r'_1$ and $r'_2$ to satisfy constraint (7). Specifically, let us adjust the value of $r_1$ be $t$ so that $r'_1 = r_1 - t$. Then, the adjustment value of $r_2$ should be at least $(\delta - \delta_i) - t$ so that $r'_2 \geq r_2 + (\delta - \delta_i) - t$. Such value adjustments indeed can satisfy constraint (7) because with the fact that $r_2 - r_1 = \delta_i$, we have

$$
|r'_2 - r'_1| \geq |r_2 - (r_1 - t)| + (\delta - \delta_i).
$$

Also, for $r'_1$ and $r'_2$ to satisfy constraint (8), suppose that $r'_2$ is adjusted for exactly the amount of $(\delta - \delta_i) - t$. 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:

$$
\mu_{i'} = \frac{1}{m} \sum_{i=1}^{m} p_{i}'' = \frac{1}{N_b + N_w} \sum_{i=1}^{N_b} p_{i}'' + \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} p_{i}''
$$

$$
= \frac{1}{N_b + N_w} \sum_{i=1}^{N_b} q_i + \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} \frac{1}{N_b + N_w} \sum_{i=N_b+1}^{N_b+N_w} q_i
$$

$$
= \frac{1}{N_b + N_w} \sum_{i=1}^{N_b+N_w} q_i = \mu_{bi}.
$$

(6)
where the fact that the new value of \( p_t'' \) in \( T'' \) is set to be \( r_1' \) if the color of \( P_t \) in \( T_t \) is black; or to be \( r_2' \) if the color of \( P_t \) in \( T_t \) is white has been used in the derivations, and the last step is based on the use of (8). Accordingly, we can get:

\[
\frac{1}{N_a + N_w} N_a t - \frac{1}{N_a + N_w} N_w - N_w (\delta - \delta_1) = 0,
\]

and

\[
\frac{1}{N_a + N_w} N_a t + \frac{1}{N_a + N_w} N_w (\delta - \delta_1) - \frac{1}{N_a + N_w} N_w t = \mu_{a_t}.
\]

(10)

from which the desired value \( t \) can be derived to be

\[
t = \left\lfloor \frac{(N_a + N_w)}{t} \right\rfloor (\delta - \delta_1).
\]

The value of \( r_1' \) and \( r_2' \) can be computed by the rule:

\[
\text{if } \delta_1 < \delta_1 \text{ then set } r_1' = r_1 - t \text{ and } r_2' = r_2 + [(\delta - \delta_1) - t].
\]

(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 \( \delta \).

**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 \( b_1b_2b_3b_4b_5b_6b_7b_8 \ldots \)

Step 3. Expand every three bits \( b_1b_2b_3 \) in \( B \) into four bits \( b_1'b_2'b_3'b_4' \) according to (1) and generate the corresponding pattern block \( T' \) according to the rules shown in Fig. 5.

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

Step 8. Replace the Y-component of \( I_t \) with \( I_t' \) 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 \( I_t'' \).

![Fig. 8. Localization and correction of perspective distortion in captured signal-rich-art code image.](image)

(a) (b)

B. Block Number Identification and Block Segmentation

To identify the unit blocks in \( I_t'' \) 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 \( N_{s_t} \) of unit blocks in the horizontal or vertical direction in \( I_t'' \) 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 \( I_t'' \) is computed by a Sobel operator [23]:

\[
g_{xy} = \left\lfloor \frac{r_{x,y+1} + 2r_{x,y} + r_{x,y-1} - (r_{x+1,y} + 2r_{x+1,y} + r_{x-1,y}) + (r_{x+1,y+1} + 2r_{x+1,y} + r_{x-1,y+1}) + (r_{x-1,y-1} + 2r_{x-1,y} + r_{x+1,y-1})}{4} \right\rfloor.
\]

(13)

Next, for each possible value \( n_t \) of \( N_{s_t} \), 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:

$$x = u \times d_j \quad \text{and} \quad y = v \times d_j,$$

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

$$AG_{nj} = \frac{1}{\lfloor L/d_j \rfloor \times \lfloor L/d_j \rfloor} \sum_{j=0}^{\lfloor L/d_j \rfloor} \sum_{i=0}^{\lfloor L/d_j \rfloor} G_{(i,y)(j,x)} \times \lfloor L/d_j \rfloor \times \lfloor L/d_j \rfloor \sum_{j=0}^{\lfloor L/d_j \rfloor} \sum_{i=0}^{\lfloor L/d_j \rfloor} G_{(i,y)(j,x)}.$$

Note that in addition to the actual value $N_5$ which will yield a large average gradient value, the values $n_j$ which are divisors of $N_5$ will yield large average gradient values as well. For this, the value $n_j$ yielding the largest average gradient value $AG_{nj}$, denoted as $AG_{n}$, is selected first and those $n_j$ yielding average gradient values $AG_{nj}$ close to $AG_{n}$ are selected as well. Then, the largest $n_j$ from these selected values of $n_j$ is taken as the desired number $N_5$ 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 $N_5$, 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 $N_5$ 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 $2 \times 2$. The number of pattern blocks in the horizontal or vertical direction in $I_p''$ is so just $N_F = N_5/2$. 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 $F_{i1}'''$, $F_{i2}'''$, $F_{i3}'''$, and $F_{i4}'''$, and the color of each $F_{i}'''$ is either black or white. Hence, we may utilize the feature of blackness to describe $F_{i}'''$; that is, if the color of the unit block $F_{i}'''$ is black, then we take the blackness feature $bf_i$ of $F_{i}'''$ to be “1”; else, to be “0.” Next, we compute the real blackness feature $bf_i$ of each unit block $F_{i}'''$ in $T_i'''$ by:

$$bf_i = \text{NB}_i/(\text{NB}_i + NW_i),$$

where $\text{NB}_i$ and $\text{NW}_i$ are the numbers of black and white pixels in unit block $F_{i}'''$, respectively. Then, the absolute distance $AD_{bk}$ of the blackness feature between $T_i'''$ and $BP_k$ can be computed as:

$$AD_{bk} = \frac{1}{4} \sum_{i=1}^{4} |bf_i - bf_k|.$$

Subsequently, the code pattern $BP_m$ with the minimum absolute distance $AD_{mk}$ 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 $AD_{bk}$ 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 $AD_{b4} = 0.1425$. 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 $b_1$, $b_2$, and $b_3$ according to the bit expansion scheme.
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.

Stage 4 — Identifying pattern blocks in the code image.
Step 3. Compute the gradient value \( g_{xy} \) of each pixel \( R_{xy} \) in \( I_t'' \) according to (13).
Step 4. For each possible value \( j \) of \( N_S \), compute the average gradient values \( AG_n \) of the pixels on each candidate splitting line according to (15).
Step 5. Select the value \( j \) yielding the largest \( AG_n \), denoted as \( AG_m \), and those \( n_j \)'s yielding \( AG_n \) close to \( AG_m \); pick the largest \( n_j \) from all the selected \( n_j \)'s for use as the desired number \( N_S \) of blocks of \( I_t'' \) in the horizontal or vertical direction; and divide \( I_t'' \) accordingly into unit blocks.

Stage 5 — Binarizing the pattern blocks to extract the message.
Step 6. Group every four mutually-connected unit blocks and denote them as \( F_n \), through \( F_n \) to form a pattern block \( T_{i, q}'' \) in \( I_t'' \).
Step 7. Extract three bits \( b_1'', b_2'', \) and \( b_3'' \) from each pattern block \( T_{i, q}'' \) by the following steps.
(A) For each unit block \( F_n \) in \( T_{i, q}'' \), compute its blackness feature \( b_{nj} \) according to (16).
(B) Computing the absolute distance \( AD_n \) of \( T_{i, q}'' \) to each of the eight possible code patterns \( BP_n \) 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 \( b_{nj}'' \) through \( b_{nj}'' \) of \( T_{i, q}'' \).
(D) Take the three bits \( b_1'', b_2'', \) and \( b_3'' \) as the recovered version of the original three message bits \( b_{1s}, b_{2s}, b_{3s} \), 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 \( h_r'' \) 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 \((\text{the number of correct bits/the total number of bits})\) using different parameters including: (1) the threshold \( \delta \) 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.

### TABLE I
**AN EXAMPLE OF CODE PATTERN RECOGNITION**

<table>
<thead>
<tr>
<th>binary code pattern</th>
<th>Corresponding 4 bits</th>
<th>Absolute distance ( AD_n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>( AD_n = (0.1+0.87+0.78+0.12)/4 = 0.4675 )</td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>( AD_n = (0.1+0.87+0.22+0.88)/4 = 0.5175 )</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>( AD_n = (0.1+0.13+0.78+0.88)/4 = 0.4725 )</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>( AD_n = (0.1+0.13+0.22+0.12)/4 = 0.1425 )</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>( AD_n = (0.9+0.87+0.78+0.88)/4 = 0.8575 )</td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>( AD_n = (0.9+0.87+0.22+0.12)/4 = 0.5275 )</td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>( AD_n = (0.9+0.13+0.78+0.12)/4 = 0.4825 )</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>( AD_n = (0.9+0.13+0.22+0.88)/4 = 0.5325 )</td>
<td></td>
</tr>
</tbody>
</table>

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 \( I_t'' \).

**Stage 3 — Identifying pattern blocks in the code image.**
Step 3. Compute the gradient value \( g_{xy} \) of each pixel \( R_{xy} \) in \( I_t'' \) according to (13).

**Stage 4 — Binarizing the pattern blocks to extract the message.**
Step 4. For each possible value \( j \) of \( N_S \), compute the average gradient values \( AG_n \) of the pixels on each candidate splitting line according to (15).
Step 5. Select the value \( j \) yielding the largest \( AG_n \), denoted as \( AG_m \), and those \( n_j \)'s yielding \( AG_n \) close to \( AG_m \); pick the largest \( n_j \) from all the selected \( n_j \)'s for use as the desired number \( N_S \) of blocks of \( I_t'' \) in the horizontal or vertical direction; and divide \( I_t'' \) accordingly into unit blocks.
Step 6. Group every four mutually-connected unit blocks and denote them as \( F_n \), through \( F_n \) to form a pattern block \( T_{i, q}'' \) in \( I_t'' \).
Step 7. Extract three bits \( b_1'', b_2'', \) and \( b_3'' \) from each pattern block \( T_{i, q}'' \) by the following steps.
(A) For each unit block \( F_n \) in \( T_{i, q}'' \), compute its blackness feature \( b_{nj} \) according to (16).
(B) Computing the absolute distance \( AD_n \) of \( T_{i, q}'' \) to each of the eight possible code patterns \( BP_n \) 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 \( b_{nj}'' \) through \( b_{nj}'' \) of \( T_{i, q}'' \).
(D) Take the three bits \( b_1'', b_2'', \) and \( b_3'' \) as the recovered version of the original three message bits \( b_{1s}, b_{2s}, b_{3s} \), respectively.
Step 8. Concatenate the extracted bits into a bit stream \( B \) and transform reversely \( B \) to get the embedded message \( M' \).
One of the parameters that influence the accuracy of the extracted message is the threshold value $\delta$ for the minimum difference between the two representative values $r_1'$ and $r_2'$. If $\delta$ 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 $\delta$, 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 $\delta$, 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 $\delta$, 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).

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.

Another parameter that influences the message extraction accuracy is the number $N_V$ of unit blocks in the horizontal or vertical direction in the created pattern or code image. The larger the value of $N_V$, 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.

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.

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 by 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 $N_2$ 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

<table>
<thead>
<tr>
<th>Target image</th>
<th>$N_2$</th>
<th>Method</th>
<th>Accuracy rate (%)</th>
<th>Recognition time (ms)</th>
<th>Amount of information (bits/image)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 11(a)</td>
<td>16</td>
<td>Proposed method</td>
<td>100</td>
<td>60</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>Lee and Tsai [15]</td>
<td>100</td>
<td>1186</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>Lee and Tsai [15]</td>
<td>96.53</td>
<td>1812</td>
<td>2048</td>
</tr>
<tr>
<td>Fig. 11(c)</td>
<td>16</td>
<td>Proposed method</td>
<td>100</td>
<td>62</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>Lee and Tsai [15]</td>
<td>98.81</td>
<td>1696</td>
<td>2048</td>
</tr>
<tr>
<td>Fig. 11(e)</td>
<td>16</td>
<td>Proposed method</td>
<td>100</td>
<td>54</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>Lee and Tsai [15]</td>
<td>100</td>
<td>1230</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>Lee and Tsai [15]</td>
<td>99.76</td>
<td>83</td>
<td>4096</td>
</tr>
</tbody>
</table>

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.
and by Lee and Tsai [15].

REFERENCES


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.