Abstract—Distortion-free data embedding is a technique which can assure that not only the secret data is correctly extracted but also the cover media is recovered without any distortion after secret data is extracted completely. Because of these advantages, this technique attracts the attention of many researchers. In this paper, a new distortion-free data embedding scheme for high dynamic range (HDR) images is proposed. By depending on Cartesian product, this scheme can obtain higher embedding capacity while maintaining the exactly identical cover image and stego image when using the tone mapping algorithms. In experimental results, the proposed scheme is superior to Yu et al.'s scheme in regard to the embedding rate—an average embedding rate of 0.1355 bpp compared with Yu et al.'s scheme (0.1270 bpp).

Index Terms—Data hiding, distortion free, high dynamic range image, high embedding rate, steganography.

1. Introduction

Today, with the rapid development of multimedia technology and computer science, most information and multimedia data are exchanged by computers and the Internet without geographic limitations at any time. This leads to the wide transmission of huge amounts of information and multimedia (i.e., digital images, text, audio, and video), which means that malicious users can easily catch important content from the Internet without authorization. Therefore, ensuring the secrecy and security of data transmission is of great importance. Recently, researchers have proposed several protection algorithms, such as cryptography\cite{1,2} and data hiding\cite{3-8}. In the cryptography, the sender will convert secret data into an unrecognizable form. Then, only an authorized receiver can reconstruct this secret data into their original form by using a secret key known only by the authorized sender and receiver. However, the meaningless form of secret data will be significantly drawn by malicious users, who may attempt to decrypt messages to get information. By contrast, data hiding is a better method to guarantee the security of data transmission by avoiding the attention of malicious attackers. In data hiding, the secret information is hidden into cover digital multimedia without arousing the attention of malicious attackers. Because of the original nature of the cover, the multimedia remains even though it conveys the secret data.

Data hiding can be divided into two basic types. The first is irreversible data hiding, which hides secret data into a cover image. Many irreversible data hiding techniques\cite{4,6} have been proposed. One primary advantage of irreversible data hiding is that high hiding capacity is transmitted through the Internet. However, a weakness in irreversible data hiding is that the cover image may be damaged and cannot be reconstructed after the secret data are extracted.

The second type of data hiding is reversible data hiding\cite{3,5} (also known as lossless data hiding and distortion free data hiding), which has a reversibility feature to deal with the weakness of irreversible data hiding. This reversibility enables the cover image to be recovered correctly after extraction of secret information. This property makes reversible data hiding useful for real-time applications in medical and military areas, where the visual quality of reconstructed image is essential.

Recently, the interest of researchers has increased significantly in high dynamic range (HDR) images, which contrast to low dynamic range (LDR) images. HDR images have become popular in several fields such as computer graphics, remote sensing, digital photography, movie, computer game, medical imaging, etc. Over last three years, some data hiding schemes\cite{7,8} in HDR images have been proposed. In 2009, Cheng and Wang\cite{7} proposed the first steganography approach for HDR images. They used a two-sided algorithm that is modified from Chang and Tseng’s scheme\cite{9} and developed their own L-sided algorithm, inspired from Zhang and Wang\cite{10}. This scheme
presents the high visual quality of a stego image when a large amount of secret data is embedded. However, image distortion is unavoidable and the original image cannot be reconstructed exactly after secret data are extracted. In 2011, Yu et al. proposed a new distortion-free data hiding technique involving HDR images encoded by the radiance red-green-blue-exponent (RGBE) format. In this scheme, the secret data are embedded by depending on the advantage of some homogeneous representations inherent in the radiance RGBE encoding format. This scheme achieves the exact identity between the tone-mapped cover image and tone-mapped stego image. However, the average embedding capacity of Yu et al.’s scheme is approximately 0.12 bpp. To further improve the embedding capacity, in this paper, we propose a distortion-free data embedding scheme that depends on the Cartesian product algorithm to try using the homogeneous representations of pixels in HDR images flexibly in data embedding. The experimental results confirm that our scheme is superior to Yu et al.’s scheme in terms of the embedding capacity while guaranteeing that the tone-mapped cover image and stego-image are identical.

The rest of the paper is organized as follows. Section 2 reviews Yu et al.’s scheme. Then, the details of our proposed scheme are demonstrated in Section 3. Section 4 illustrates experimental results. Finally, some conclusions are given in Section 5.

2. Related Work

In 1991, Ward introduced the format of HDR images, also known as the 32-bit radiance RGBE format, since then which has found widespread utilization in graphics communication. In the HDR image encoded with the radiance RGBE format, the pixel value is represented by three primary channels and one exponent channel. Each channel is in the range of 0 to 255. This means that each channel will use 8 bits to store its value. Fig. 1 shows the HDR pixel encoded in the 32-bit radiance RGBE format.

Basically, the color of the HDR pixel is a floating point value. Therefore, it can be derived by using the floating point conversion which is defined as

\[
R = \left(\frac{r + 0.5}{256}\right) \times 2^{e-128}
\]

(1a)

\[
G = \left(\frac{g + 0.5}{256}\right) \times 2^{e-128}
\]

(1b)

\[
B = \left(\frac{b + 0.5}{256}\right) \times 2^{e-128}
\]

(1c)

where the color pixel \((R, G, B)\) represents the floating point. Assume that \(P(r, g, b, e)\) indicates the pixel encoded with the radiance RGBE format, with the three primary color channels \(r, g, b\) and \(e\) shown as an exponent channel, respectively.

Likewise, the color pixel with the floating point \((R, G, B)\) can be converted into the radiance RGBE format \((r, g, b, e)\) by using integer conversion as shown in (2):

\[
e = \log_2\left[\max\left(R,\frac{G}{2^{e-128}},\frac{B}{2^{e-128}}\right) + 128\right]
\]

\[
r = \left[\frac{256 \times R}{2^{128}}\right]
\]

\[
g = \left[\frac{256 \times G}{2^{128}}\right]
\]

\[
b = \left[\frac{256 \times B}{2^{128}}\right]
\]

(2d)

It is clear that with the exponent channel \(e\) in the radiance format more than one representation can be used to show one color pixel. By using the division operation with the divisor 2 for each primary channel and adding 1 to the exponent channel, or by using the multiplication operation with the multiplier 2 for each primary channel and subtracting 1 from the exponent channel, we can get the new representation of pixels that can provide nearly the same floating point color value and also give the identical color value with the original pixel when a tone mapping scheme is applied. For example, assume that an original pixel is \(P(r, g, b, e)\). By using division or multiplication, the new representations are produced as \(P(r/2, g/2, b/2, e+1)\) or \(P(rx2, gx2, bx2, e-1)\), respectively. However, the new representation \(P(rx2, gx2, bx2, e-1)\) must satisfy that each channel still has an integer value. For the new representation \(P(rx2, gx2, bx2, e-1)\), each channel must be in the legal range from 0 to 255. By using the above mentioned process, each pixel contains a number of different representations. In other words, each pixel has a set of different representations, known as the homogeneous representation group (RG). In the RG group, each element can describe the same color pixel as the original one \(P(r, g, b, e)\). We define the number of elements in the group RG as \(N\). Table 1 shows the detailed example of the sorted RG group with the number of elements \(N\).

By exploiting the RG group of each pixel in an HDR image, Yu et al., in 2011, proposed a new distortion-free data hiding algorithm that can embed secret messages into the HDR images. The advantage of Yu et al.’s scheme is that it can convey the secret message to generate the stego HDR image. Moreover, when the tone mapping technique is processed, there is no distortion between the original and stego images. In this scheme, a RG group of pixels, as shown in Table 1, is exploited to embed the secret bits. The homogeneous index table is generated in advance to support embedding data, as shown in Table 2.

![Fig. 1. HDR pixel in 32-Bit of radiance RGBE format.](image)

<table>
<thead>
<tr>
<th>Table 1: Detail example of RG group with (N=3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color pixel (P)</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>((26,72,132,128))</td>
</tr>
<tr>
<td>((26,72,132,128))</td>
</tr>
<tr>
<td>((13,36,63,129))</td>
</tr>
</tbody>
</table>
The embedding algorithm is shown as follows.

Input: the HDR cover image $I$, homogeneous index table HIT, and secret message $B$.

Output: HDR stego image $I'$. 

Step 1. Read a pixel $P$ from the image $I$.

Step 2. Determine the sorted homogeneous representation group $RG_P$ and the number of elements $N$ of group $RG_P$.

Step 3. If $N\leq 1$, it means that no secret bits can be embedded, go back to Step 1. Otherwise, compute the number of bits of secret bits $b_P$ that can be extracted by using (3).

Step 4. Determine the suitable homogeneity index $d$ of stego pixel $P'$ from the group $RG_P$.

Step 5. By depending on $N$ and the homogeneity index $d$, the extracted secret bits $b_P$ is found from the table HIT. Then, $b_P$ is sent to the secret message $B$.

Step 6. Repeat Steps 1 to 5 until all pixels in image $I'$ are processed completely.

After the above steps are processed completely, the secret message $B$ is extracted. Here is an example to explain our extracting phase in detail. Suppose that the stego pixel $P'$ is $(16, 12, 40, 129)$. Now, to extract the secret bits $b_P$ from the pixel $P'$, the group $RG_P$ of $P'$ is determined as Table 3, and the number of element $N$ of group $RG_P$ equals 5. It is easy to see that the homogeneity index of pixel $P'$ is $d=2$ in the group $RG_P$. Therefore, through $N$ and the homogeneity index $d$, secret bits $b_P$ are found in the HIT table, which are the extracted secret bits. Then, $b_P$ is sent to the message $B$.

### 3. Proposed Scheme

The proposed scheme is described in detail in this section. The cover image is the HDR image $I$ sized $W\times H$, and the secret message $B$ is denoted as $B=(b_0, b_1, \cdots, b_j)$, where $b_j$ is the secret bit generated randomly and $b_j \in \{0, 1\}$, $0\leq j \leq S$. The proposed distortion-free data hiding scheme can be divided into two phases, data embedding and data extracting, which are discussed in Subsection 3.1 and Subsection 3.2, respectively.

### 3.1 Data Embedding Phase

After carefully observing Yu et al.’s hiding scheme[8], we figure out that for each pixel, the number of secret bits is $\lceil \log_2(N) \rceil$, which is embedded into each pixel of image $I$, where $N$ is the number of elements of RG group. Therefore, the total embedding capacity of image $I$ sized $W\times H$ is

$$\text{capacity} = \sum_{i=1}^{W\times H} \lceil \log_2(N_i) \rceil.$$  (4)

### Table 3: Homogeneous representation group $RG_P$ of color pixel $P(32, 24, 80, 128)$

<table>
<thead>
<tr>
<th>Color pixel $P$</th>
<th>Number of elements $N$ in group $RG_P$</th>
<th>Homogeneity index</th>
<th>Sorted elements of in group $RG_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(32, 24, 80, 128)$</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>(64, 48, 160, 127)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>(32, 24, 80, 128)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(16, 12, 40, 129)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(8, 6, 20, 130)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(4, 3, 10, 131)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It is obvious to see that the summary of $\lfloor \log_2 a \rfloor$ and $\lfloor \log_2 b \rfloor$ will be smaller or equal to $\lfloor \log_2 (ab) \rfloor$, where $a$ and $b$ are integer values. For example, if $a=5$ and $b=13$, respectively, $\lfloor \log_2 5 \rfloor = \lfloor \log_2 13 \rfloor = 3 \leq \lfloor \log_2 (5 \times 13) \rfloor = \lfloor \log_2 65 \rfloor = 6$. Therefore, by exploiting the above mentioned idea, in this paper, we use a Cartesian product algorithm to further improve the embedding capacity of Yu et al.’s scheme while guaranteeing the high quality of the stego image.

Basically, a Cartesian product is the direct product of two sets $A$ and $B$, which is denoted as $A \times B$ and $A$ and $B$ are the ordered set of all possible ordered pairs whose first component is a member of $A$ and whose second component is a member of $B$, as given by

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B \}.$$ (5)

For example, the Cartesian product of 4-element set $A$, namely by \{1, 2, 3, 4\}, and 3-element set $B$ given as \{x, y, z\}, is the 12-element set with all possible ordered pairs as \{(1, x), (1, y), (1, z), (2, x), \ldots, (4, z)\}. The corresponding Cartesian product has $4 \times 3 = 12$ elements.

In general, a Cartesian product of two finite sets can be shown by a table, with one set as the rows and the others as the columns and creating the order pairs, the cells of the table, by selecting the set from the row and column.

Fig. 2 shows the flowchart of our embedding algorithm, which involves four operations, namely determining the RG group for each pixel, computing the Cartesian product $S$ of all RG groups, then selecting the suitable bit string for each element in $S$, and the embedding process.

Our proposed data embedding phase can be divided into four steps. The corresponding algorithm is shown in detail as below.

**Input:** the original HDR image $I$ sized $W \times H$ and secret message $B$.

**Output:** the stego image $I'$.

**Step 1.** For each pixel $P_i$, determine the sorted homogeneous representation group $RG_i$ and the corresponding number of element $N_i$ in $RG_i$, where $i$ is in the range from 1 to $W \times H$.

**Step 2.** Compute the Cartesian product $S$ of all groups $RG_i$ by using (6), where the number of elements of set $S$, $|S|$, is computed by using (7).

$$S = RG_i \times RG_{i+1} \times \cdots \times RG_{W \times H}$$ (6)

$$|S| = N_i \times N_{i+1} \times \cdots \times N_{W \times H}.$$ (7)

**Step 3.** Compute the suitable secret bit string $r_i$ for each element of set $S$. Here, the length of $r_i$ is calculated by

$$\text{length of } r_i = \lfloor \log_2 |S| \rfloor = \lfloor \log_2 (N_i \times N_{i+1} \times \cdots \times N_{W \times H}) \rfloor.$$ (8)

**Step 4.** To embed the secret message $B$, the corresponding element in $S$, which has secret bit string $r_i$ equal to $B$, is found. Then this element is sent out as the stego image $I'$.

![Detailed flowchart of embedding algorithm](image)

Fig. 2. Detailed flowchart of embedding algorithm.

![Original image and RG groups](image)

Fig. 3. Original image $I$ and correspond $RG_i$ and $N_i$ of each pixel.

![Stego image](image)

Fig. 4. Stego image $I'$.

After these four steps are processed completely, the stego image $I'$ is obtained, which is sent to the receiver without any extra information. This example explains the data embedding phase in detail. Assume that the original HDR image is given as Fig. 3.

In Fig. 3, it is clear to see that the first pixel $P_1$ has group $RG_1 = \{x_1, y_1\}$ and its number of elements of group $RG_1, N_1 = 2$. Then the Cartesian product $S$ of the 4 groups $RG_1, RG_2, RG_3, RG_4$ is 36 possible elements as $\{(x_1, x_2, x_3, x_4), (x_1, x_2, x_3, y_4), \ldots, (y_1, z_2, y_3, z_4)\}$. According to (8), the number of elements of set $S$ will be $|S| = 2 \times 3 \times 2 \times 3 = 36$ elements. For each element in set $S$, we match one secret bit string $r_i$ which has the length $\lfloor \log_2 |S| \rfloor = 5$ bits (from “00000” to “11111”). This means that we can embed the secret message $B$ which has the same length as the secret bit string $r_i$. Therefore, suppose that the secret message $B = “00010”$ is embedded. According to Step 4 in the embedding algorithm, we can get the stego image $I'(x_1, x_2, x_3, z_4)$, which is the element in the set $S$, have the secret bit string $r_i$ that is the same as the secret message $B$. Then, the stego image $I'$ is generated as presented in Fig. 4.
3.2 Data Extracting Phase

After receiving the stego image $I'$ from the sender, the receiver can extract the secret message $B$ exactly by using the following steps in our data extracting phase. A detailed flowchart of our data extracting phase is provided in Fig. 5.

Extracting algorithm:
Input: the stego image $I'$.
Output: the extracted secret message $B$.

Step 1. For each pixel $P_i$ from the image $I'$, determine the sorted homogeneous representation group $RG_i$ and the corresponding number of element $N_i$ of group $RG_i$, where $i$ is in the range from 1 to $W\times H$.

Step 2. Compute the Cartesian product $S$ of all groups $RG_i$ by using (6), where the number of element of set $S$, $|S|$, is determined by (7).

Step 3. Select the suitable secret bit string $r_i$ for each element of set $S$. Here, the length of $r_i$ is calculated by (8).

Step 4. To extract the secret message $B$, find the element in the set $S$ which has the same color value as the stego image $I'$. Then the corresponding secret bit string of this element is the extracted secret message $B$.

After these four steps, the secret message $B$ is extracted correctly. For example, suppose that the stego HDR image $I'$ is given, as shown in Fig. 6. Then determine the corresponding group $RG_i$ and the number of elements $N_i$ for each pixel of the stego image, as shown in Fig. 6. The Cartesian product $S$ of 4 groups $RG_1$, $RG_2$, $RG_3$, and $RG_4$ has 36 possible elements as $\{(x_1, y_1, z_1), \ldots, (y_1, z_2, y_3, z_4)\}$, and the number of element of set $S$ is $|S|=24\times 3\times 2\times 3=36$ elements. For each element in $S$, the corresponding secret bit string $r_i$ has the length $\lceil \log_2|S|\rceil=\lceil \log_2 36 \rceil=5$ bits (from “00000” to “11111”). Then, to extract the secret message, the set $S$ is searched to find the element which has the same value as the stego image $I'$. When the match element is found, the secret message $B$ is extracted as the corresponding secret bit string $r_i$ of the match element. Thus, for this case, the stego image $I'$ equals to the third element $(x_1, x_2, x_3, x_4)$. Therefore, the extracted secret message $B$ is the corresponding secret bit string $r_i$ of the third element, “00110”.

4. Experimental Results

To illustrate the performance of our proposed scheme and Yu et al.'s schemes[8], five HDR test images, “Church,” “Hall,” “Aspen,” “Bush,” and “Pine” presented in Fig. 7 were used in our experiments. The size of the first four images is 720x480 pixels, and that of the last test image is 2000x1312 pixels. All computing was performed on a PC with a 2.1 GHz Intel(R) Core™2 CPU and a 1 GB RAM. The operating system was Windows 7 Professional and our algorithm was programmed by Microsoft Visual Studio 2005 C#.

Table 4 shows the details of the five HDR test images. For example, the “Church” HDR image has 302482 pixels which have the number of element $N=1$ in the group $RG_1$, and has 37640 pixels when $N=2$, respectively.

![Fig. 5. Detailed flowchart of our data extracting phase.](image5)

![Fig. 6. Example of extracting phase.](image6)

![Fig. 7. Five HDR test images: (a) Church, (b) Hall, (c) Aspen, (d) Bush, and (e) Pine.](image7)
Table 4: Detail characteristics of five HDR test images

<table>
<thead>
<tr>
<th>Test images (pixels)</th>
<th>Number of elements $N$ in group RG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Church</td>
<td>302482</td>
</tr>
<tr>
<td>Hall</td>
<td>302411</td>
</tr>
<tr>
<td>Aspen</td>
<td>302036</td>
</tr>
<tr>
<td>Bush</td>
<td>302876</td>
</tr>
<tr>
<td>Pine</td>
<td>2294522</td>
</tr>
</tbody>
</table>

Table 5: Embedding capacity of our scheme and Yu et al.’s scheme\[^8\] for five HDR test images

<table>
<thead>
<tr>
<th>Methods</th>
<th>720×480</th>
<th>2000×1312</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Church</td>
<td>Hall</td>
<td>Aspen</td>
</tr>
<tr>
<td>Total capacity (pixels)</td>
<td>46710</td>
<td>46868</td>
<td>47218</td>
</tr>
<tr>
<td>Embedding rate (bpp)</td>
<td>0.1352</td>
<td>0.1356</td>
<td>0.1366</td>
</tr>
<tr>
<td>Yu et al.’s scheme</td>
<td>43810</td>
<td>43839</td>
<td>44284</td>
</tr>
<tr>
<td>Total capacity (pixels)</td>
<td>0.1268</td>
<td>0.1268</td>
<td>0.1281</td>
</tr>
<tr>
<td>Embedding rate (bpp)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Tone mapped cover and stego of three test images: (a) the tone mapped cover image “Hall” and the tone mapped stego image “Hall”, (b) the tone mapped cover image “Aspen” and the tone mapped stego image “Aspen”, and (c) the tone mapped cover image “Bush” and the tone mapped stego image “Bush”.

Table 5 presents the comparison of our scheme and Yu et al.’s scheme\[^8\] in term of the embedding capacity. Obviously, the embedding capacity of our scheme is better in all HDR images. The average embedding rate of our scheme is 0.1355 bpp, and that of Yu et al.’s scheme is 0.1270 bpp. Yu et al.’s scheme does not use some homogeneous representation to embed secret bits. For example, in line 7 of Table 2, when the number of elements in the group RG equals 6, Yu et al.’s scheme only used the first four homogeneous representations to embed two secret bits, and the last two representations have not been used. Besides, in Yu et al.’s scheme, the homogeneous index table HIT is needed as extra information that is sent to the receiver to support the extracting process. In contrast, our scheme applies the Cartesian product to try to use all homogeneous representations of each group RG, and no extra information is required. Thus, our scheme can achieve higher embedding capacity compared with Yu et al.’s scheme.

Fig. 8 provides the tone mapping results of utilizing the Luminance HDR software for the HDR images “Hall,” “Aspen,” and “Bush.” Obviously, the tone mapped cover images are identical with different types of tone mapping algorithms. When the Reinhard et al.’s tone mapping scheme\[^12\] is used with the parameters of gamma=1.000, brightness=−10.0, chromatic=1.00, and light=0.00, respectively, the tone mapped cover image “Hall” and the tone mapped stego image “Hall” are identical, as shown in Fig. 8 (a). Fig. 8 (b) presents the tone mapping result of the HDR image “Aspen” by utilizing the tone mapping method of Durand and Dorsey\[^13\] with four parameters gamma=1.000, spatial=8.00, range=0.40, and contrast=5.00, respectively. The “Bush” HDR image using the tone mapping scheme of Mantuik et al.\[^14\] is shown in Fig. 8 (c), and three parameters, gamma, contrast, and saturation, are set as 1.000, 0.300, and 0.800, respectively. From Fig. 8 the tone mapped cover image and tone mapped stego image are identical; therefore, we can conclude that our proposed scheme leads to a distortion-free data hiding scheme for HDR images encoded in the RGBE format.

5. Conclusions

In this paper, the distortion-free data hiding scheme for HDR image encoded in the 32-bits radiance RGBE format is proposed to further improve the performances of Yu et al.’s scheme\[^8\]. In term of the embedding capacity, the average embedding rate of our proposed scheme is around 0.1355 bpp, which outperforms Yu et al.’s scheme (0.1270 bpp). This is because, in our scheme, all homogeneous representations are used to embed secret bits. Moreover, the experiment shows that the tone mapped cover and stego images are identical. It means that our scheme tends to distortion-free data embedding. In other words, when compared with Yu et al.’s scheme, our scheme achieves a better result in embedding capacity, while maintaining the good quality of the stego image. However, the embedding capacity is still low, for the average embedding capacity is smaller than 0.15 bpp. Thus, in the future, we intend to design a new distortion free data embedding algorithm for HDR images with higher embedding capacity by using all homogeneous representations of each pixel more efficiently.
References


Chin-Chen Chang was born in Taichung, Taiwan in 1954. He received his B.Sc. degree in applied mathematics in 1977, M.Sc. degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, and Ph.D. degree in computer engineering in 1982 from the National Chiao Tung University, Hsinchu. During the academic years of 1980 to 1983, he was with the Department of Computer Engineering, National Chiao Tung University. From 1983 to 1989, he was with the Institute of Applied Mathematics, National Chung Cheng University, Chiayi; from August 1992 to July 1995, he was the Dean of the College of Engineering at the same university. From August 1995 to October 1997, he was the Provost of National Chung Cheng University. From July 1998 to June 2000, he was the Director of Advisory Office of the Ministry of Education of Taiwan. From 2002 to 2005, he was a chair professor with National Chung Cheng University. Since February 2005, he has been a chair professor with Feng Chia University. In addition, he has served as a consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures. Dr. Chang is a fellow of the IEEE.

Thai-Son Nguyen received the bachelor’s degree in information technology from the Open University, Ho Chi Minh City in 2005. In 2011, He received his M.S. degree in computer sciences from Feng Chia University, Tai Chung. From December 2006, he has been a lecturer with Travinh University, Tra Vinh. His current research interests include data hiding, signal processing, image compression, multimedia security, and information security.

Chia-Chen Lin received the B. S. degree in information management from the Tamkang University, Taipei, Taiwan in 1992. She received the M.S. and the Ph.D. degrees both in information management from Chiao Tung University, Hsinchu in 1994 and 1998, respectively. She was a visiting associate professor at Business School, University Illinois at Urbana Champain, during August 2006 to July 2007. She is currently a professor with the Department of Computer and Information Management, Providence University, Sha-Lu. Her research interests include image and signal processing, image data hiding, mobile agent, and electronic commerce.