

A new distortion-free data embedding scheme for high-dynamic range images

Chin-Chen Chang · Thai-Son Nguyen · Chia-Chen Lin

Received: 3 February 2014 / Revised: 14 July 2014 / Accepted: 15 September 2014

Published online: 28 September 2014

© Springer Science+Business Media New York 2014

Abstract Distortion-free data embedding enables a cover image to be recovered from its stego-image without any distortion after the secret data have been extracted. This distortion-free property allows the appropriate recovery of highly sensitive images for which even the smallest modification of the cover images cannot be allowed. In this paper, a new distortion-free data embedding scheme is proposed for high-dynamic range (HDR) images. The proposed scheme uses all homogeneous representations of each pixel in an HDR image efficiently and effectively for data embedding to enhance the embedding capacity of the HDR cover image. First, in the embedding phase, all homogeneous representations are used, and each homogeneous representation is used to represent one pattern of secret bits. Then, to conceal the secret bits, the current homogeneous representation of the current processing pixel is modified by the corresponding homogeneous representation of the hidden secret bits. Experimental results confirmed that the proposed scheme has greater embedding capacity than three other existing schemes. In addition, the experimental results indicated that our scheme matched the visual quality of the stego-image by producing a tone-mapped cover image and its stego-image that were exactly the same. In other words, our scheme also provided the desired, distortion-free property.

Keywords Data embedding · Distortion-free · High-dynamic range image · Steganography · High embedding rate

C.-C. Chang · T.-S. Nguyen

Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724
Taiwan, Republic of China

C.-C. Chang
e-mail: ccc@cs.ccu.edu.tw

T.-S. Nguyen
e-mail: thaison@tvu.edu.vn

C.-C. Lin (✉)

Department of Computer Science and Information Management, Providence University, No. 200 Chung-chi
Rd., Taichung 43301 Taiwan, Republic of China
e-mail: mhlin3@pu.edu.tw

1 Introduction

Data embedding is a very important component of making sure that information is secure. In data embedding, the secret information is embedded into cover digital media, such as images, videos, audios, or texts, before it is transmitted over a public channel, e.g., the Internet. Data embedding ensures that the original form of the cover media is maintained even though the cover media contain the secret message. This makes it difficult for malicious attackers to determine whether media contain secret data or not, thereby guaranteeing the security of the embedded secret data.

In principle, data embedding schemes designed for images can be classified into three domains. The first domain consists of spatial domain schemes [1, 4, 6, 18, 24, 30], in which each pixel value of a cover image is modified to embed the secret data. The second domain is the transform domain [16, 28] in which the cover image is converted into coefficient values by using various transform techniques, such as the integer discrete cosine transform [16] and the integer wavelet transform [28]. Then, these coefficients are modified to embed the secret data. The third domain is the compression domain [2, 3, 11, 15] in which the cover image is compressed by some popular techniques, such as vector quantization [2, 15], block truncation coding [3], and JPEG coding [11], after which the compressed codes of an image are processed to hide the secret message.

Data embedding also can be divided into two basic types based on reversibility. The first type is irreversible data embedding [1, 18, 24, 25]. By using irreversible data embedding, a large amount of secret data can be hidden and transmitted through popular channels at the cost of not being able to recover the cover images. The second type is reversible data embedding [2, 22, 27], which is also called distortion-free data embedding or lossless data embedding. Distortion-free data embedding was proposed to overcome the weakness of irreversible data embedding in that it allows the original cover image to be recovered after the hidden secret data have been extracted. This property makes distortion-free data embedding useful for some sensitive applications, such as medical and military applications, for which the cover images must be fully restored after the extraction of the hidden secret data.

In the last decade, many scholars have designed irreversible data embedding schemes for low-dynamic range (LDR) images. In contrast with LDR images, high-dynamic range (HDR) images have a greater dynamic range of luminance between the lightest and darkest areas of an image. With their wide dynamic range, HDR images can represent the range of intensity levels found in real scenes, from direct sunlight to the deepest shadows, more accurately than LDR images. This capability of HDR images has attracted the attention of many researchers in several fields, such as movie/computer games, computer graphics, remote sensing, digital photography, and medical images. Over the last 3 years, many data embedding schemes [2–30] for high-dynamic range (HDR) images have been presented. In 2009, Cheng and Wang [8] proposed an adaptive steganography approach for HDR images. To embed secret data, they used a two-sided algorithm that was modified from that used in Chang and Tseng's scheme [6]. Then, they proposed their own L-sided algorithm, which was inspired by Zhang and Wang's work [30]. Cheng and Wang's scheme provides high embedding capacity, in the range 5.13–9.69 bits out of 32 bits of RGBE encoding. However, the visual quality of the original image is diminished, and the original cover image cannot be reconstructed exactly after the embedded secret data are extracted. To improve the embedding rate of Cheng and Wang's scheme, Li et al. [14] proposed a new data hiding scheme for HDR images. Instead of using HDR images encoded in 32-bit RGBE coding, Li et al.'s scheme used 48-bit TIFF coding of HDR images and the LSB substitution technique for embedding data. By doing that, Li et al.'s scheme achieves a high embedding rate, since each pixel can carry seven secret bits. However,

by using 48-bit TIFF coding of HDR images, their scheme significantly increased the size of the original image. In addition, their scheme cannot reconstruct the original HDR image after the data have been extracted. To the best of our knowledge, in 2011, Yu et al. [29] were the first to introduce a distortion-free, data embedding technique for HDR images, which was encoded by the radiance RGBE format [26]. In their scheme for embedding secret data, only partial, homogeneous representations of each pixel in the radiance RGBE encoding format were used to generate a homogeneous representation group, and this approach has been used extensively in the graphics community. Their scheme can obtain an embedding capacity of about 0.12 bpp. In addition, the scheme maintains good visual quality of the stego-image, because the tone-mapped cover image and the tone-mapped stego-image are exactly the same. However, in their scheme, partial homogeneous representations in the homogeneous representation group are excluded when the data are being embedded. Hence, the embedding capacity of Yu et al.'s scheme is less than 0.13 bpp. To further improve the performance of Yu et al.'s scheme, in 2012, Wang et al. [23] introduced a segment-based data hiding scheme. In their scheme, some neighboring pixels are segmented into groups. Then, each group is used to embed a decimal digit. By doing that, all homogeneous representations are used flexibly. In 2013, Solachidis et al. [14] introduced a watermark scheme for HDR images based on bracketing decomposition. Their scheme obtains high detection rates of attacks while ensuring the imperceptibility of the embedded watermark. Also in 2013, Chang et al. [5] proposed a new, distortion-free data hiding scheme for HDR images. In their scheme, the Cartesian product is applied to try to use all of the homogeneous representations of the HDR pixels in a practical manner. Their scheme obtains a better embedding rate than Yu et al.'s scheme, but the embedding capacity of their scheme is still low, i.e., less than 0.14 bpp. In this paper, we propose a new, distortion-free data embedding scheme that uses all homogeneous representations of the HDR images encoded with the radiance RGBE format [26] to provide greater embedding capacity than three existing schemes [5, 23, 29]. The experimental results confirmed that our scheme provides a greater embedding capacity than these three existing schemes. In addition, the identities of the tone-mapped cover image and the stego-image generated by our scheme are guaranteed to be the same.

The rest of the paper is organized as follows. A review of Yu et al.'s scheme [29] is presented in Section 2, and the details of our proposed scheme are provided in Section 3. The experimental results are presented in Section 4, and our conclusions are presented in Section 5.

2 Yu et al.'s scheme

The format of HDR images, also known as the 32-bit radiance RGBE format, was first introduced by Ward [26] in 1991. Since then, HDR images have become increasingly popular in graphic communications. Basically, the pixel value in the HDR image encoded in the radiance RGBE format consists of four components, i.e., three primary components (r , g , b) and one exponential component, e . Thus, $P(r, g, b, e)$ is used to denote an HDR pixel. Each component of the HDR pixel uses eight bits to present its value. Therefore, the value of each component is in the range of 0 to 255. Figure 1 shows an HDR pixel encoded in the radiance RGBE format.

A pixel value, $P(r, g, b, e)$, in the radiance RGBE format can be converted from a color pixel with a floating point (R, G, B) by using integer conversion, as shown in Eq (1).



Fig. 1 HDR pixel encoded in the 32-bit radiance RGBE format

$$\begin{aligned}
 e &= \lceil \log_2[\max(R, G, B)] + 128 \rceil \\
 r &= \left\lceil \frac{(256 \times R)}{2^{e-128}} \right\rceil \\
 g &= \left\lceil \frac{(256 \times G)}{2^{e-128}} \right\rceil \\
 b &= \left\lceil \frac{(256 \times B)}{2^{e-128}} \right\rceil
 \end{aligned}
 \tag{1}$$

where (r, g, b) are three primary components, e is one exponential component of the pixel value in the radiance RGBE format, and (R, G, B) are the three components of the color pixel., () is round bracket, and [] denotes the square bracket.

There is a special case for an input color pixel with the floating point (R, G, B) in which the maximum (R, G, B) is less than 10⁻³⁸. In this case, the pixel value, P(r, g, b, e), can be written as (0, 0, 0, 0), which also is known as the zero pixel.

Similarly, a color pixel with a floating point (R, G, B) can be derived by using the floating point conversion, which is shown in Eq (2):

$$\begin{aligned}
 R &= \left((r + 0.5) / 256 \right) \times 2^{(e-128)} \\
 G &= \left((g + 0.5) / 256 \right) \times 2^{(e-128)} \\
 B &= \left((b + 0.5) / 256 \right) \times 2^{(e-128)}
 \end{aligned}
 \tag{2}$$

where color pixel (R, G, B) represents the floating point, and P(r, g, b, e) indicates the pixel encoded with radiance RGBE format; r, g, and b are the three primary color components, and e is the exponent component.

Obviously, the exponent component, e, used in the radiance RGBE format, allows the color pixel to have more than one representation. For example, assume that the original pixel is P(r, g, b, e). We can divide each primary component by two and add one to the exponent component. Then, the new representation can be generated as D(r/2, g/2, b/2, e+1) by using division and addition operations. Similarly, we also can multiply each primary component by two and subtract one from the exponent component. By doing this, the new representation of the pixel would be M(r*2, g*2, b*2, e-1). After the above processes, many homogeneous representations of one pixel can be derived, providing an exactly identical color after tone-mapping. However, for the new representation D(r/2, g/2, b/2, e+1), the value of each primary component is required in integer form, and, for M(r*2, g*2, b*2, e-1), the value of each primary component is required in the range between 0 and 255. By using the division and multiplication operations, a set of different homogeneous representations is produced that also is referred to as the homogeneous representation group (HRG). Each element of the HRG group can show the same color value as the original element. The number of elements in the HRG group is defined as the homogeneity value N. Examples of sorted homogeneous representation HDR groups are provided in Table 1.

In 2011, Yu et al. [29] proposed the first distortion-free data embedding in an HDR image encoded with the radiance RGBE format. To embed secret data, first, Yu et al.'s scheme determines all homogeneous representations of each pixel. A homogeneity index table (HIT),

Table 1 HDR groups and the homogeneity value N of pixel (34, 18, 124, 128)

Color Pixel P	Homogeneity value N	Homogeneity index	Sorted homogeneous representation group, HRG
P(34, 18, 124, 128)	3	0	(68,36,248,127) (34,18,124,128)
		1	(17,9,62,129)
		2	

as shown in Table 2, is generated in advance to provide support for both the data embedding and data extracting phases.

During the data embedding phase, the cover pixel $P(r, g, b, e)$ can have secret bits embedded based on the HIT . When a pixel P is read from a cover image, the homogeneity value N of P is determined. The first column of Table 2 lists the number of embedded bits $|b|$, which is computed according to homogeneity value N , as shown in Eq (3):

$$|b| = \lfloor \log_2(N) \rfloor. \tag{3}$$

The third column of Table 2 shows the bit pattern of the secret message, which is hidden with respect to different indices. By using the HIT , secret bits can be embedded into each pixel. The following example is provided to illustrate the process and further clarify Yu et al.’s data embedding process. Assume that an original pixel $P(20, 16, 60, 127)$ is given. Hence, to embed secret data, the sorted homogeneous group HRG_P and homogeneity value N are determined, as shown in Table 3. By utilizing Eq (3), the number of secret bits that can be embedded into pixel P can be determined, and it is $|b|=2$. Suppose that secret bits b are read as “01.” Based on homogeneity value $N=5$ and secret bits $b=“01,”$ the homogeneity index value $HI=0$ is selected from the HIT table. To hide two secret bits, i.e., “01,” the stego-pixel is provided as an output element of the homogeneous representation group HGR_P . This element must have the same homogeneity index, i.e., $HI=0$. In this case, the stego-pixel becomes $P'(80, 64, 240, 125)$ in Table 3. In another example, if the secret bits $b=“00”$ are to be embedded, the homogeneity index $HI=3$ would be chosen based on the homogeneity value $N=5$ from the HIT . Finally, the stego-pixel P' is changed to the element in the homogeneous representation group, which has the same homogeneity index. In other words, the pixel $P'(10, 8, 30, 128)$ is sent out as the stego-pixel.

Table 2 Homogeneity index table (HIT)

Numbers of embedded bits $ b $	Homogeneity value N	Homogeneity index							
		0	1	2	3	4	5	6	7
0	1	NP	–	–	–	–	–	–	–
1	2	“0”	“1”	–	–	–	–	–	–
1	3	“1”	“0”	NA	–	–	–	–	–
2	4	“00”	“01”	“10”	“11”	–	–	–	–
2	5	“01”	“10”	“11”	“00”	NA	–	–	–
2	6	“10”	“11”	“00”	“01”	NA	NA	–	–
2	7	“11”	“00”	“01”	“10”	NA	NA	NA	–

Table 3 Sorted homogeneous representation group HRG_P of pixel P (20, 16, 60, 127)

Color Pixel P	Homogeneity value N	Homogeneity index	Sorted homogeneous representation group, HRG_P
P (20, 16, 60, 127)	5	0	(80, 64, 240, 125)
		1	(40, 32, 120, 126)
		2	(20, 16, 60, 127)
		3	(10, 8, 30, 128)
		4	(5, 4, 15, 129)

3 Proposed scheme

In this section, our proposed scheme is described in detail. Here, we assume that the cover image is an HDR image I sized $W \times H$ and that the secret message B with size of k is defined as $B = (b_0, b_1, \dots, b_k)$, where b_j is the secret bit generated randomly, i.e., $b_j = \{0, 1\}$ and $0 \leq j \leq k$. The proposed scheme is broken into two phases, i.e., the data embedding phase and the data extracting phase, which are discussed in Subsections 3.1 and 3.2, respectively.

3.1 Data embedding phase

After we deeply explored Yu et al.’s hiding scheme [29], we realized that it did not use the homogeneous representation group (HRG) efficiently. The total embedding capacity of Yu et al.’s scheme with a cover image I with the size of $W \times H$ is computed as shown in Eq (4):

$$capacity = \sum_{i=1}^{W \times H} \lfloor \log_2(N_i) \rfloor, \tag{4}$$

where N is the number of elements in the HRG group. The number of embedded secret data of each pixel is $\lfloor \log_2(N) \rfloor$ bits. Hence, partial homogenous representations in the homogeneous representation group are not used to carry secret bits. In this paper, we proposed an improved, distortion-free data embedding scheme to enhance the embedding capacity of Yu et al.’s scheme while maintaining the high quality of the stego-image.

In our data embedding scheme, the homogeneity index table is created in advance for use during the data embedding and data extracting phases. Table 4 shows our homogeneity index table (HIT). Comparing Table 2 generated by Yu et al.’s scheme with Table 4 generated by our scheme, it is apparent that only the first four indices are used to embed two secret bits when the homogeneity value, N , is equal to 6 in Yu et al.’s scheme. However, in our proposed scheme, all homogeneity indices are used to carry two or three secret bits.

Figure 2 shows a flowchart of our data embedding phase. It contains three operations for each pixel, i.e., generating the homogeneity index table, determining the sorted homogeneous representation group HRG , and the data embedding operation.

To determine the sorted homogeneous representation group HRG , the division and multiplication operations are used to find all the homogeneous representations of each pixel, just as Yu et al.’s scheme does. Then, all of the homogeneous representations that are found are set as the homogeneous representation group. The homogeneous representation group is sorted by the exponent component. In addition, the homogeneity value N is determined as the number of elements of the sorted homogeneous representation group HRG .

As for the second operation, i.e., generating the homogeneous index table, the homogeneous value N means that there are N homogeneity indices, which are labeled from 0 to $N-1$,

Table 4 Our homogeneity index table (*HIT*) that is used for both the embedding and extracting processes

Number of embedded bits $ b $	Homogeneity value N	Homogeneity index						
		0	1	2	3	4	5	6
0	1	NA	–	–	–	–	–	–
1	2	“1”	“0”	–	–	–	–	–
1 or 2	3	“1”	“01”	“00”	–	–	–	–
2	4	“11”	“10”	“01”	“00”	–	–	–
2 or 3	5	“11”	“10”	“01”	“001”	“000”	–	–
2 or 3	6	“11”	“10”	“011”	“010”	“001”	“000”	–
2 or 3	7	“11”	“101”	“100”	“011”	“010”	“001”	“000”

and the homogeneous value N must be between 1 and 7. Then, the binary tree of N homogeneity indices is created. The pattern of secret bits of each homogeneity index is set up based on the binary tree. The algorithm for generating the homogeneous index table is provided below, and Fig. 3 shows an example of a binary tree and the pattern of the secret bits after the algorithm to better explain it.

- Step 1 If homogeneous value $N=1$, the binary tree cannot be created, meaning that this is a non-embeddable case and is not applicable (NA) for embedding secret bits. Therefore, the homogeneity index 0 of homogeneous index is marked as NA.
- Step 2 Otherwise (i.e., $N \neq 1$), it is an embeddable case and the binary tree can be created.
- Step 3 Create set $M = \{M_1, M_2, \dots, M_N\}$ of N nodes, and all elements of set M are leaf nodes of the binary tree. Assign a node M_i with the homogeneity index i , where M_i is labeled as the name of the node, and i is set as the value of node.
- Step 4 Sort the N nodes according to the value of the homogeneity index i in ascending order. From the N nodes, select the two nodes that have the largest values and generate the new node as the parent of both nodes. When a new parent node is created, it is labeled

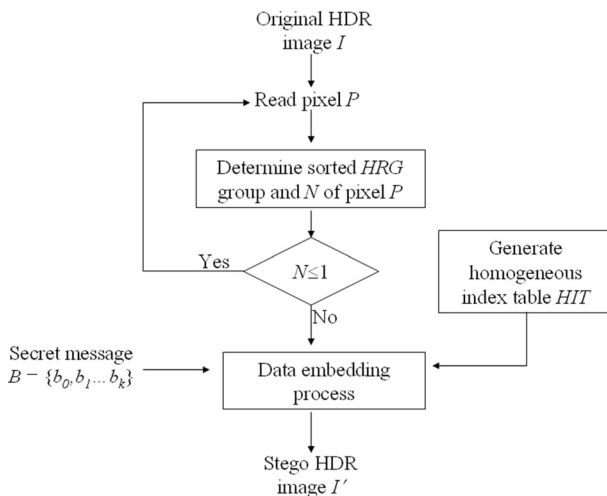


Fig. 2 Flowchart of data embedding phase

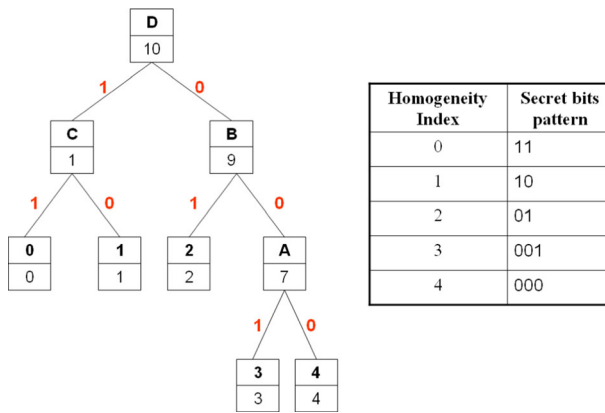


Fig. 3 Binary tree and pattern of the secret bits for each corresponding homogeneity index

with a character (e.g., A, B, C,...), and the value of the parent node’s homogeneous index is set as the sum of the values of two child nodes. Note that the larger node is the right child node, and the smaller node is the left child node.

- Step 5 Set the branch value between the left child node and its parent node as “1,” and set the branch value between the right child node and its parent node as “0.”
- Step 6 Set M is updated by replacing the two child nodes with their newly-generated parent node.
- Step 7 If the number of elements in set M is greater than 1, go to Step 4. Otherwise, go to Step 8.
- Step 8 Traverse from the root of the generated binary tree to each leaf node M_i to create the pattern of the secret bits of the corresponding homogeneity index i , by concatenating the branch value (1 or 0) from the root to the leaf node.

Repeat the eight steps listed above until all of the homogeneity values, i.e., the N_s , are processed completely to derive the pattern of the secret bits of each homogeneity index of homogeneity value N in the range of [1, 8]. To give a clear explanation, an example of a binary tree is provided for a homogeneity value, N , of 5. In this case, the homogeneity index ranges from 0 to 4. After using the above eight steps, a binary tree and the pattern of the secret bits of each homogeneity index can be generated, as shown in Fig. 3.

Our proposed data embedding algorithm can be broken down into six steps, which are shown in detail below.

Data embedding algorithm:

Input: Original HDR cover image I sized $w \times h$, secret message B , homogeneity index table HIT

Output: Stego-image I'

Step 1: Read pixel P from original HDR cover image I .

Step 2: Determine the sorted homogeneous representation group HRG and homogeneity value N of pixel P

Step 3: If $N \leq 1$, pixel P belongs to a non-embedding case, meaning that no secret bits are hidden in this case.

Step 4: Otherwise, read secret bits b . Depending on secret bits b and homogenous value N , the corresponding homogeneity index d is selected from the HIT table.

Step 5: Based on the selected index d , choose the corresponding homogeneous representation, which is the stego-pixel P' of pixel P Finally, stego-pixel P' is output to stego-image I' .

Step 6: Repeat Steps 1 through 5 until all pixels of image I have been processed.

After the data embedding algorithm is completed, the stego-image I' is obtained. Finally, stego-image I' is sent to the receiver. To give a better explanation of the data embedding algorithm, the following example is given. Assume an original HDR pixel P is (20, 16, 60, 127). The homogeneity value $N=5$ of pixel P and its homogeneous representation group HRG are determined based on the division and multiplication operations, and they are sorted depending on the exponent component, which is shown in Table 3. According to the proposed HIT shown in Table 4, two or three secret bits can be embedded in each index. Assume that the secret bits b are “001,” and the corresponding homogeneity index $d=3$ is selected. Finally, we choose the corresponding homogeneous representation of index d in homogeneous representation group HRG , i.e., (10, 8, 30, 128), which is the stego-pixel P' . Then, P' is in the RGBE format, and it is sent to stego-image I' .

3.2 Data extracting phase

Once the receiver receives stego-image I' and homogeneity index table HIT from the sender, he or she extracts secret message B exactly by utilizing our data-extracting phase, the flowchart for which is shown in Fig. 4. It contains three operations for each pixel, i.e., generating the homogeneity index table, determining the sorted homogeneous representation group HRG , and the data extracting operation. The first and second operations are the same as those in the data embedding phase; therefore, they are not repeated here

The data extracting algorithm is presented below.

Data extracting algorithm:

Input: Stego-image I' , homogeneity index table HIT

Output: Extracted secret message B

Step 1: Read stego-pixel P' from stego-image I' .

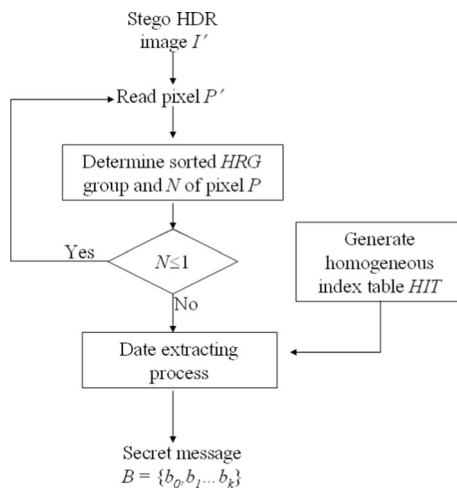


Fig. 4 Flowchart of our data extracting phase

Step 2: Determine the sorted homogeneous representation group HRG and homogeneity value N of stego-pixel P' based on the division operation and the multiplication operation.

Step 3: If $N \leq 1$, no secret bits are extracted from the current processing of stego-pixel P' . Otherwise, go to Step 4.

Step 4: Based on the value of stego-pixel P' , select the corresponding homogeneity index d from the homogeneous representation group HRG .

Step 5: Depending on the index d and homogeneity value N , the extracted secret bits b are determined from the HIT table. Then, b is sent to secret message B .

Step 6: Repeat Steps 1 through 5 until all pixels of image I' are processed.

After the above data extracting algorithm is processed completely, the secret message B is extracted correctly. To describe the data extracting phase in detail, the following example is a continuation of the example given above to illustrate the data embedding phase. Assume that the stego-pixel P' (10, 8, 30, 128) is given. First, the homogeneity index table HIT is generated by using the same operation of the data embedding phase. The homogeneous representation group HRG of pixel P' is determined by using the division and multiplication operations. Then, this homogeneous representation group is sorted based on the exponent component, and homogeneity value $N=5$ is determined as the number of the sorted homogeneous representation group's (HRG 's) elements, as shown in Table 3. According to Table 3, the homogeneity index d that corresponds to P' is three. By using homogeneity index d and homogeneity value N , we can determine the extracted secret bits $b="001"$ in the homogeneity index table HIT . Then, the extracted secret bits b are concatenated to secret message B .

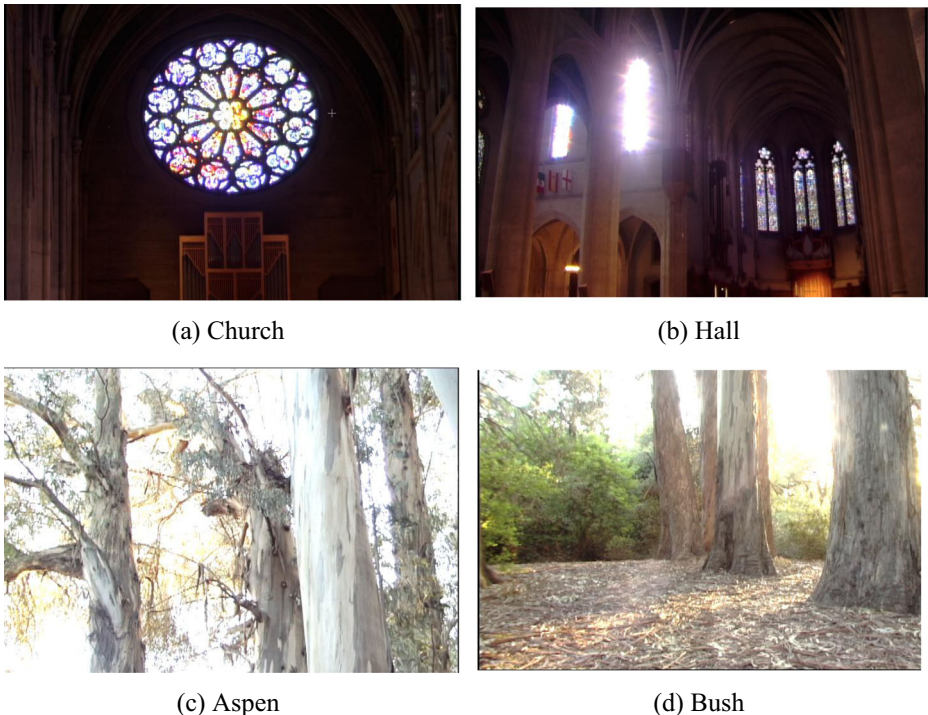


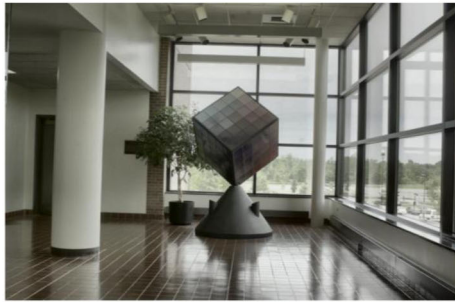
Fig. 5 Ten HDR test images. **a** Church, **b** Hall, **c** Aspen, **d** Bush, **e** Pine, **f** Clay, **g** Cube, **h** Park, **i** Plant, **j** Nancy



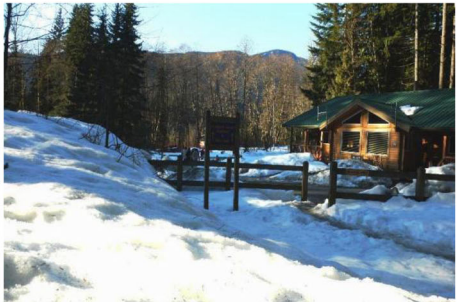
(e) Pine



(f) Clay



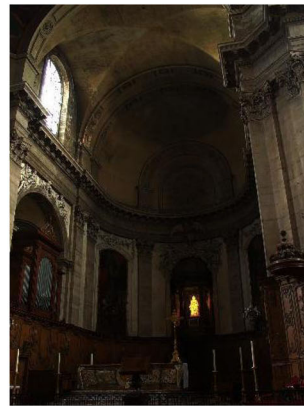
(g) Cube



(h) Park



(i) Plant



(j) Nancy

Fig. 5 (continued)

4 Experimental results

In this section, the hiding capacity performance of our proposed scheme is compared with that of four existing schemes, i.e., Yu et al.'s scheme [29], Wang et al.'s scheme [23], Chang et al.'s scheme [5], and Li et al.'s scheme [14]. Ten HDR test images, shown in Fig. 5, were used in our experiments. Notably, the first four test images, i.e., images (a) – (d), were 720×480 pixels in size, and two of the test images, i.e.,

images (e) and (g), were $2,000 \times 1,312$ pixels, whereas test image (f) was $3,840 \times 2,160$ pixels, and test image (h) was $2,048 \times 1,536$ pixels. Secret message B was generated using a pseudo-random number generator. These 10 HDR test images were formatted in the 32-bit RGBE coding and considered as input images in the proposed scheme, as well as in Yu et al.'s scheme [29], Wang et al.'s scheme [23], and Chang et al.'s scheme [5]. In Wang et al.'s scheme, each input image is segmented into 1,000 groups for embedding the secret data. In Li et al.'s scheme, the 10 HDR test images are converted to 48-bit TIFF coding that are considered as the input images for embedding the secret data. These settings ensured that each of the previous schemes could provide its best performance in order to make a fair comparison of our proposed scheme with their schemes. The experimental setup included a PC with an Intel(R) Core™ 2 CPU @ 2.1-GHz and a 2-GB RAM. The operating system used for the tests was Windows 7 Professional 32-bit. In this paper, all of the algorithms were implemented by MATLAB R2010a.

To give the reader an understanding of the number of embeddable and non-embeddable cases regarding the test images presented above, the number of pixels with different homogeneity values in each image is demonstrated in Table 5. For example, the “Church” HDR image has 302,482 pixels that have a homogeneity value equal to 1. However, these pixels contain a special pixel, known as the zero pixel, that has three primary components and an exponent component, all of which are zeros. This means that secret bits cannot be embedded in zero pixels because, when pixel P is a zero pixel, we can divide 255 times to generate the homogeneous representation group with the homogeneity value of $N=256$, such as $\{(0,0,0,0), (0,0,0,1), \dots, (0,0,0,255)\}$. By doing this, we can embed eight bits for this pixel. However, an unexpected pixel difference could occur, which would degrade the visual quality of the image because of the repeated application of the division operation. The reason is that 0.5 was added to the floating point conversion in Eq (2). In addition, the floor function was used in the integer conversion in Eq (1). Therefore, in our scheme, zero pixels are not used to embed secret bits to avoid having a visual difference between the tone-mapped cover image and the tone-mapped stego-image when the tone-mapping schemes are used. Another example is that, in the “Hall” image, there are a total of 2,397 zero pixels that are not used to embed secret bits.

Table 5 Detailed characteristics of the five HDR test images

Test images	Homogeneity values (N)						
	1	2	3	4	5	6	7
Church	302,482 (2618)	37,640	4,786	600	82	8	2
Hall	302,411 (2397)	37,540	4,999	558	83	9	0
Aspen	302,036 (2779)	38,021	4,823	606	92	21	1
Bush	302,876 (2807)	37,405	4,645	587	74	12	1
Pine	2,294,522 (0)	287,488	36,178	4,776	1,008	27	1
Plant	2,297,199 (0)	286,745	34,396	5,113	482	62	3
Clay	2,290,174 (0)	286,867	36,969	4,359	585	48	3
Cube	2,289,761 (0)	287,038	36,517	4,701	527	48	66
Nancy	2,746,982 (0)	349,570	42,853	5,533	643	147	0
Park	7,252,915 (0)	910,910	114,190	14,248	1,895	237	5

Also, the “Hall” image contains 37,540 pixels, has a homogeneity value of $N=2$, and has 558 pixels that have homogeneity values of three, as shown in Table 5.

Table 6 shows that the embedding capacity (EC) and embedding rate (ER) performance of the proposed scheme, Yu et al.’s scheme [29], Wang et al.’s scheme [23], Chang et al.’s scheme [5], and Li et al.’s scheme [14]. Table 6 shows that Li et al.’s scheme provided the highest embedding rate among the five schemes. This is because their scheme used the LSB substitution technique for embedding secret data. As a consequence, their scheme cannot reconstruct the original image after the data have been extracted. Obviously, for the four reversible data hiding schemes, the embedding capacity of the proposed scheme is superior to that of the other three schemes. The average embedding capacity of the proposed scheme is 0.1445 bpp. By contrast, the average embedding capacities of Chang et al.’s scheme, Wang et al.’s scheme, and Yu et al.’s scheme are only 0.1355 bpp, 0.1337 bpp, and 0.1273 bpp, respectively. The embedding rate of our scheme is better because all homogeneous representations in our scheme are used to embed secret bits. In contrast, Yu et al.’s scheme has the worst embedding rate of the four schemes. In Yu et al.’s scheme, only some first homogeneous representations are used to embed secret bits, because once the homogeneity value is N , the number of embedded secret data of each pixel is $\lfloor \log_2(N) \rfloor$ bits. Thus, some of the last homogeneous representations in the homogeneous representation group are not used to carry secret bits. For example, in line 8 of Table 2, when the homogeneous value $N=6$, only the first four homogeneous representations are used to embed two secret bits in Yu et al.’s scheme. However, all homogeneous representations are used in our scheme to embed two or three secret bits, as shown in line 8 of Table 4.

In Fig. 6, tone-mapping results are presented in which the Luminance HDR software was used for the three HDR images “Hall,” “Aspen,” and “Bush.” For the “Hall” images, Reinhard et al.’s tone-mapping method [19] was used for “Hall” with four parameters, i.e., $\gamma = 1.000$, $\text{brightness} = -10.0$, $\text{chromatic} = 1.00$, and $\text{light} = 0.00$. Figure 6a shows the exact identity of the tone-mapped cover image “Hall” and the tone-mapped stego-image “Hall.” The tone-mapping results of HDR image “Aspen” are presented in Fig. 6b with the tone-mapping method of Durand and Dorsey [10] and four parameters, i.e., $\gamma = 1.000$, $\text{spatial} = 8.00$, $\text{range} = 0.40$, and $\text{contrast} = 5.00$. In Fig. 6c, the tone-mapping scheme of Mantiuk et al. [17] was used for the “Bush” image, which has three parameters, i.e., γ , contrast , and saturation , which were set as 1.000, 0.300, and 0.800, respectively. Obviously, in Fig. 6, the exact identity of the tone-mapped cover image and the tone-mapped stego-image are presented. In addition, this result can be derived from Eq (2), in which the exponent component e , used in the radiance RGBE format, allows the color pixel to have more than one representation. For example, the original pixel, $P(r, g, b, e)$, can be shown by the new representation as $D(r/2, g/2, b/2, e+1)$ by using division and addition operations. Similarly, each primary component can be multiplied by two, and the exponent component is subtracted to one to generate a new representation of the pixel as $M(r*2, g*2, b*2, e-1)$. Since e is the exponent component, to guarantee that the color of the pixel with floating point (R, G, B) in Eq (2) remains unchanged, three primary components must be divided by 2 when the value of e is increased by 1 in $D(r/2, g/2, b/2, e+1)$. In contrast, to guarantee that the color pixel with floating point (R, G, B) in Eq (2) remains unchanged, three primary components must be multiplied by 2 when the value of e is decreased by 1 in $M(r*2, g*2, b*2, e-1)$. By doing that, many homogeneous representations can be determined with the original pixel being exactly the same color. Thus, using the division and multiplication operations, a set of different homogeneous representations is produced, and each homogeneous representation is used for indicating for a pattern of secret bits in the proposed scheme. As a result, the proposed scheme can embed the secret bits by using one element of the set of different homogeneous

Table 6 Comparison of embedding capacity of our scheme and four existing schemes

Test images Schemes	480×720				1,312×2,000				1,536×2,048				3,840×2,160
	Church	Hall	Aspen	Bush	Pine	Plant	Clay	Cube	Nancy	Park	Average		
Proposed scheme	Reversible	EC (bit/s)	49,377	49,299	48,063	386,436	382,582	384,635	386,129	459,745	1,213,437	340,843	
		ER (bpp)	0.1410	0.1429	0.1426	0.1391	0.1473	0.1458	0.1466	0.1472	0.1461	0.1463	
Yú et al.'s scheme	Reversible	EC (bit/s)	43,839	44,284	43,398	335,290	332,461	333,826	334,239	405,069	1,057,870	297,409	
		ER (bpp)	0.1268	0.1268	0.1281	0.1256	0.1278	0.1267	0.1272	0.1274	0.1288	0.1275	
Wáng et al.'s scheme	Reversible	EC (bit/s)	45,377	45,942	44,889	355,468	351,548	353,126	354,283	431,878	1,121,377	314,934	
		ER (bpp)	0.1315	0.1313	0.1329	0.1299	0.1355	0.1340	0.1346	0.1350	0.1373	0.1352	
Chang et al.'s scheme	Reversible	EC (bit/s)	46,868	47,218	46,215	357,316	352,340	353,892	354,916	433,120	1,124,521	316,312	
		ER (bpp)	0.1352	0.1356	0.1366	0.1337	0.1362	0.1343	0.1349	0.1353	0.1377	0.1356	
Li et al.'s scheme	Irreversible	ER (bpp)	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	7,000	



(a) Tone-mapped cover image “Hall” and tone-mapped stego-image “Hall”



(b) Tone-mapped cover image “Aspen” and tone-mapped stego-image “Aspen”



(c) Tone-mapped cover image “Bush” and tone-mapped stego-image “Bush”

Fig. 6 Tone-mapped cover images and corresponding tone-mapped stego-images. **a** Tone-mapped cover image “Hall” and tone-mapped stego-image “Hall”. **b** Tone-mapped cover image “Aspen” and tone-mapped stego-image “Aspen”. **c** Tone-mapped cover image “Bush” and tone-mapped stego-image “Bush”

representations generated from the original pixel. Therefore, we can confirm that our proposed scheme provides a distortion-free, data embedding technique for HDR images that are encoded in the 32-bit RGBE format.

Table 7 compares the average execution times of our proposed scheme and three existing schemes. Obviously, Wang et al.’s scheme requires the highest average execution time (3.068 s) in both the embedding and extracting phases, followed by Chang et al.’s scheme (2.647 s), Yu et al.’s scheme (2.264 s), our proposed scheme (2.264 s), and Li et al.’s scheme (1.237 s). Wang et al.’s scheme is the worst one among the four schemes. This is because Wang et al.’s scheme takes much more time than others for dividing the image into groups of pixels and embedding secret decimal digits into each group separately. Li et al.’s scheme has the lowest execution time,

Table 7 Comparisons of the average execution times of our scheme and three existing schemes (unit: second)

Schemes	Embedding time, unit	Extracting time, unit
Proposed scheme	1.291	0.973
Yu et al.'s scheme	1.291	0.973
Wang et al.'s scheme	1.786	1.282
Chang et al.'s scheme	1.475	1.172
Li et al.'s scheme	0.886	0.351

because their scheme applied LSB substitution for embedding secret data, and it does not take time for searching the homogeneity representations of each pixel. However, Li et al.'s scheme is an irreversible data hiding scheme. Yu et al.'s scheme and our proposed scheme showed the best performance in saving processing time. The two schemes provided the same results, because they were based on the same characteristics of the HDR image used to embed the secret data. However, our proposed scheme provided better embedding capacity.

5 Conclusions

In this paper, we proposed a new, distortion-free, data-embedding scheme for an HDR image encoded in the 32-bit radiance RGBE format. In comparison, with three other reversible data hiding schemes [5, 23, 29] for HDR images, our scheme achieved the best performance. The average embedding rate of our scheme was around 0.1445 bpp, which is superior to the 0.1355 bpp provided by Chang et al.'s scheme, the 0.1337 bpp provided by Wang et al.'s scheme, and the 0.1273 bpp provided by Yu et al.'s scheme. This is because all homogeneous representations are used to embed secret bits in our scheme. In addition, in the experiment, the tone-mapped cover image and the tone-mapped stego-image were exactly identical, which is similar to Yu et al.'s scheme. In other words, our scheme is a distortion-free, data-embedding technique. Moreover, the proposed scheme reduces the execution time as does Yu et al.'s scheme. Therefore, due to our scheme's high performances mentioned above, it can be concluded that our proposed scheme is sufficiently efficient and flexible to apply directly in some applications, such as digital libraries or online multi-media transmissions.

References

1. Chan CK, Cheng LM (2004) Hiding data in images by simple LSB substitution. *Pattern Recogn* 37:469–474
2. Chang CC, Lin CY (2007) Reversible steganographic scheme using SMVQ approach based on declustering. *Inf Sci* 177(8):1796–1805
3. Chang CC, Lin CY, Fan YH (2008) Lossless data hiding for color images based on block truncation coding. *Pattern Recogn* 41(7):2347–2357

4. Chang CC, Lu TC (2006) A difference expansion oriented data hiding scheme for restoring the original host images. *J Syst Softw* 79(12):1754–1766
5. Chang CC, Nguyen TS, Lin CC (2013) Distortion-free data embedding scheme for high dynamic range images. *J Electron Sci Technol* 11(1):20–26
6. Chang CC, Tseng HW (2004) A steganographic method for digital images using side match. *Pattern Recogn Lett* 25(14):1431–1437
7. Chao MW, Lin CH, Yu CW, Lee TY (2009) A high capacity 3D steganography algorithm. *IEEE Trans Vis Comput Graph* 15(2):274–284
8. Cheng YM, Wang CM (2009) A novel approach to steganography in high-dynamic-range images. *IEEE MultiMedia* 16(3):70–80
9. Drago F, Myszkowski K, Annen T, Chiba N (2003) Adaptive logarithmic mapping for displaying high contrast scenes. *Comput Graph Forum* 22(3):419–426, Proceedings of Eurographics 2003
10. Durand F, Dorsey J (2002) Fast bilateral filtering for display of high-dynamic-range images. *Proc ACM SIGGRAPH 2002*:257–266
11. Fridrich J, Goljan M, Du R (2001) Invertible authentication watermark for JPEG images, Proceedings of IEEE International Conference on Information Technology: Coding and Computing, Las Vegas, NV, USA, 223–227
12. Huang NC, Li MT, Wang CM (2009) Toward optimal embedding capacity for permutation steganography. *IEEE Signal Process Lett* 16(9):802–805
13. Larson GW, Shakespeare RA (1988) *Rendering with radiance*, San Francisco, Morgan Kaufmann
14. Li MT, Huang NC, Wang CM (2011) A data hiding scheme for high dynamic range images, *Int J Innov Comput Inf Control* 7(5)
15. Lin CY, Chang CC (2006) Hiding data in VQ-compressed images using dissimilar pairs. *J Comput* 17(2):3–10
16. Lin CC, Shiu PF (2010) DCT-based reversible data hiding scheme. *J Softw* 5(2):214–224
17. Mantiuk R, Myszkowski K, Seidel HP (2006) A perceptual framework for contrast processing of high dynamic range images. *ACM Trans Appl Percept* 3(3):286–308
18. Mielikainen J (2006) LSB matching revisited. *IEEE Signal Process Lett* 13(5):285–287
19. Reinhard E, Pattanaik S, Ward G, Debevec P (2005) *High dynamic range imaging, Acquisition, Display, and Image-Based Lighting*, San Francisco, Morgan Kaufmann
20. Reinhard E, Stark M, Shirley P, Ferwerda J (2002) Photographic tone reproduction for digital images. *ACM Trans Graph* 21(3):267–276
21. Solachidis V, Maiorana E, Campisi P, Banterle F (2013) HDR image watermarking based on bracketing decomposition, 18th International Conference on Digital Signal Processing (DSP), Nomikos Conference Centre, Santorini, Greece
22. Tian J (2003) Reversible data embedding using difference expansion. *IEEE Trans Circ Syst Video Technol* 13(8):890–896
23. Wang ZH, Chang CC, Lin TY, Lin CC A novel distortion-free data hiding scheme for high dynamic range images, Proceedings of the 2012 Fourth International Conference on Digital Home, 33–38
24. Wang RZ, Lin CF, Lin JC (2001) Image hiding by optimal LSB substitution and genetic algorithm. *Pattern Recogn* 34:671–683
25. Wang CM, Wu NI, Tsai CS, Hwang MS (2008) A high quality steganographic method with pixel-value differencing and modulus function. *J Syst Software* 81(1):150–158
26. Ward G (1991) Real pixel, *Graphic Gem II*, 80–83 (Chapter 15)
27. Wu HC, Wang HC, Tsai CS (2010) Reversible image steganographic scheme via predictive coding. *Displays* 31(1):35–43
28. Xuan GR, Yang CY, Zhen YZ, Shi YQ, Ni ZC (2004) Reversible data hiding using integer wavelet transform and companding technique, Proceedings of the Third International Workshop on Digital Watermarking, Seoul, Korea, 115–124
29. Yu CM, Wu KC, Wang CM (2011) Distortion-free data hiding scheme for high dynamic range images, *Displays*
30. Zhang X, Wang S (2005) Steganographic using multiple-base notational system and human vision sensitivity. *IEEE Signal Process Lett* 12(1):67–70



Chin-Chen Chang (M'88–SM'92–F'99) received the B.S. degree in applied mathematics and the M.S. degree in computer and decision sciences from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1977 and 1979, respectively. He received the Ph.D degree in computer engineering from National Chiao Tung University, Hsinchu, in 1982.

During the academic years 1980–1983, he was on the faculty of the Department of Computer Engineering, National Chiao Tung University. From 1983 to 1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From August 1989 to July 1992, he was Head of, and a Professor in, the Institute of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan. 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 September 1996 to October 1997, he was Acting President of National Chung Cheng University. From July 1998 to June 2000, he was Director of the Advisory Office, Ministry of Education, R.O.C. From 2002 to 2005, he was a Chair Professor at National Chung Cheng University. From February 2005, he has been a Chair Professor at Feng Chia University. In addition, he was severd as a Consultant to several research institutes and government departments. His current research interests include database design, computer cryptography, image compression, and data structures.



Thai-Son Nguyen received the bachelor's degree in information technology from the Open University, HCM city, Vietnam, in 2005. From December 2006 to August 2009, he was a lecturer of Travinh University, Travinh, Vietnam. Since September 2009, he has been with the MSN lab, Feng Chia University, Tai Chung, Taiwan. His current research interests include data hiding, image processing, image compression, and multimedia security.



Chia-Chen Lin (M'08) received the B. S. degree in information management from the Tamkang University, Taipei, Taiwan, R.O.C., in 1992. She received the M.S. degree in information management and the Ph.D. degree in information management from Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1998, respectively. She was a Visiting Associate Professor at Business School, University Illinois at Urbana Champaign, during August 2006 to July 2007. She is currently the associate editor of *Journal of Information Hiding and Multimedia Signal Processing* and the vice chair of Tainan Chapter, IEEE Signal Processing Society. Now, she is a professor in the Department of Computer and Information Management, Providence University, Sha-Lu, Taiwan. Her research interests include image and signal processing, image data hiding, mobile agent, and electronic commerce.