Two-in-One Image Steganography Using Error Diffusion

Dong, Ruixi
Department of Communication Design Science, Graduate School of Design, Kyushu University

Inoue, Kohei
Department of Communication Design Science, Graduate School of Design, Kyushu University

Hara, Kenji
Department of Communication Design Science, Graduate School of Design, Kyushu University

Urahama, Kiichi
Department of Communication Design Science, Graduate School of Design, Kyushu University

https://hdl.handle.net/2324/2235204

出版情報：Journal of the Institute of Industrial Applications Engineers. 7 (2), pp.42-50, 2019-04-25. 産業応用工学会
バージョン：
権利関係：Creative Commons Attribution 4.0 International (CC BY 4.0)
Two-in-One Image Steganography Using Error Diffusion

RUIXI DONG* Non-member, KOHEI INOUE* Member
KENJI HARA* Non-member, KIICHI URAHAMA* Member

(Received December 6, 2018, revised April 9, 2019)

Abstract: Image steganography is a technique for concealing a secret message in a cover image unobtrusively. The resultant images are called the stego images. In this paper, we propose a method for concealing a secret image into a cover image of the same size, where the most significant bits (MSBs) of the secret image are embedded in the least significant bits (LSBs) of the cover image after the reversal of the order of the bit sequences. Such a symmetric relationship between MSBs and LSBs derives a complementary between the stego and extracted secret images. We also propose a method for improving the image quality of both stego and extracted secret images by using an error diffusion technique. Experimental results show that the proposed method works well for both grayscale and color images, and the proposed error diffusion method can suppress the noises like false contours caused in the embedding process visually and quantitatively.

Keywords: Image steganography, Most significant bit, Least significant bit, Error diffusion

1. Introduction

The developments of information technology in recent years have demanded secure communication among the information technology equipments and the users. Hiding information, such as copyright messages and serial numbers, is a promising technique for information security, and has recently become important in a number of application areas [1]. Information hiding techniques include two subdisciplines: watermarking and steganography [2]. In watermarking, hidden information in a carrier signal should have a relationship with the carrier signal, e.g., watermarking can be applied to ownership assertion, transaction tracking and content authentication of the carrier signal [3]. On the other hand, in steganography, hidden information has no relationship with the carrier signal generally, and the presence of the hidden information is hidden [4].

Steganalysis is a counterpart of steganography, and has been extensively studied in the last decade [5]. Xia et al. proposed an improved version of Gabor filter residual (GFR) steganalysis [6]. Qian et al. proposed a paradigm for steganalysis to learn features automatically via deep learning models [5]. Agarwal and Farid detected manipulations such as insertion, removal, rotation and airbrushing from JPEG dimples [7].

Digital images are one of the potential candidates for the carrier signal in information hiding, because digital images have high redundancy, where we can embed secret messages, and pervasive applications in daily life [8]. Therefore, image steganography has lately attracted much attention from researchers, and a number of standard methods for image steganography have been presented thus far [9]. Among spatial domain steganography [2], the least significant bit (LSB) steganography [10] is a well-known approach, and has a number of its variants such as the enhanced LSB steganography [11] and the modified LSB algorithm [12]. Hadidi and Ibrahim proposed a 4-LSB method which uses 4 LSBs of 24-bit true color image for hiding text message [13]. Baluja proposed the deep steganography which attempts to place a full size color image within another image of the same size with deep neural networks [14].

The above image steganography methods conceal secret messages including images to produce stego images. As a result, the produced stego images are changed from the original cover images, i.e., the cover images are corrupted in the process of embedding secret messages. This observation motivated us to alleviate the error in the stego images using an image processing technique.

In this paper, we propose a two-in-one image steganography method which conceals a secret image into a cover image of the same size in an LSB steganography approach, where 4 most significant bits (MSBs) of the secret image are embedded in the corresponding 4 LSBs of the cover image after the reversal of the order of the bit sequence of the 4 MSBs. Such a bitwise operations cause the change of pixel values, which may be noticeable visually. To alleviate the noticeable error between the original and the changed pixel values, we introduce an error diffusion technique, which improves the image quality of the stego and extracted secret images. Experimental results show that the proposed method can conceal a secret image into a cover image for both grayscale and color images, and the image quality of the stego and extracted secret images is improved by the proposed error diffusion method.

The rest of this paper is organized as follows: Section 2

* Corresponding: k-inoue@design.kyushu-u.ac.jp
Department of Communication Design Science, Kyushu University
4-9-1, Shiobaru, Minami-ku, Fukuoka 815-8540, Japan
proposes a two-in-one image steganography method for grayscale and color images. Section 3 shows experimental results. Section 4 discusses the results. Finally, Section 5 concludes this paper.

2. Proposed Two-in-One Image Steganography

In this section, we first describe our two-in-one image steganography method for grayscale images, and then describe that for color images.

2.1 Grayscale Image Steganography Let $F = \{f_{ij}\}$ and $G = \{g_{ij}\}$ be two grayscale images, where $f_{ij}$ and $g_{ij}$ denote the pixel values at the position $(i, j)$ in $F$ and $G$, respectively, for $(i, j) \in \Omega$ where $\Omega = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ where $\times$ denotes the Cartesian product of two sets, and $m$ and $n$ denote the numbers of rows and columns, respectively. The procedure of the proposed grayscale image steganography is divided into two parts: bitwise operations and error diffusion as follows.

2.1.1 Bitwise Operations Assume that $F$ and $G$ are cover and secret images, respectively, and $f_{ij}$ and $g_{ij}$ are expressed in the decimal system, i.e., $f_{ij} \in \{0, 1, \ldots, 255\}$ and $g_{ij} \in \{0, 1, \ldots, 255\}$ for 8-bit images. To show explicitly that $f_{ij}$ is a decimal number, we use the expression $(f_{ij})_{10}$. Then we convert $(f_{ij})_{10}$ into the corresponding binary number $(b_{1:4}^F b_{5:8}^F)_{2}$, where $b_{k}^F \in \{0, 1\}$ for $k = 1, 2, \ldots, 8$. That is, $f_{ij}$ can be computed from $\{b_{1}^F, b_{2}^F, \ldots, b_{8}^F\}$ by $f_{ij} = b_{1}^F \times 2^7 + b_{2}^F \times 2^6 + b_{3}^F \times 2^5 + b_{4}^F \times 2^4 + b_{5}^F \times 2^3 + b_{6}^F \times 2^2 + b_{7}^F \times 2^1 + b_{8}^F \times 2^0$. For notational convenience, we introduce an abbreviation as follows: $(b_{1:4}^F b_{5:8}^F)_{2}$.

We also convert $(g_{ij})_{10}$ into the binary number $(b_{1:4}^G b_{5:8}^G)_{2}$ in the same way as $(f_{ij})_{10}$ as shown in Figure 1. Next, we extract 4 most significant bits (MSBs) from both $(b_{1:4}^F b_{5:8}^F)_{2}$ and $(b_{1:4}^G b_{5:8}^G)_{2}$, i.e., we take $b_{1:4}^F$ and $b_{1:4}^G$ as denoted in green and blue, respectively, in Figure 1. Then we reverse the order of the latter $b_{1:4}^G$ as $b_{1:4}^{G'} = b_{1}^{G'} b_{2}^{G'} b_{3}^{G'} b_{4}^{G'}$, and combine it with $b_{1:4}^F$ as $(b_{1:4}^F b_{1:4}^{G'})_{2}$ which is converted into the decimal number as $(h_{ij}^F)_{10} = (b_{1:4}^F b_{1:4}^{G'})_{2}$ as shown in Figure 1. This procedure is performed for all pixels. As a result, we obtain the stego image $H^F = \{h_{ij}^F\}$ which appears to be similar to $F$, but conceals a partial information of $G$ in it.

\[
\begin{align*}
\text{Decimal} & \quad \text{Binary} \\
\text{Image } F & \quad f_{ij} \quad b_{1}^F b_{2}^F b_{3}^F b_{4}^F b_{5}^G b_{6}^G b_{7}^G b_{8}^G \\
\text{Image } G & \quad g_{ij} \quad b_{1}^G b_{2}^G b_{3}^G b_{4}^G b_{5}^F b_{6}^F b_{7}^F b_{8}^F \\
\text{Image } H^F & \quad h_{ij}^F \quad b_{1}^F b_{2}^F b_{3}^F b_{4}^F b_{5}^G b_{6}^G b_{7}^G b_{8}^G \\
\text{Image } H^G & \quad h_{ij}^G \quad b_{1}^F b_{2}^F b_{3}^F b_{4}^F b_{5}^G b_{6}^G b_{7}^G b_{8}^G
\end{align*}
\]

Figure 1: Conversion between decimal and binary numbers of pixel values.

The procedure for extracting the concealed secret image from the stego image $H^F$ is as follows: First, we convert the decimal pixel value $(h_{ij}^F)_{10}$ of $H^F$ into the binary number $(b_{1:4}^F b_{5:8}^F)_{2}$, and then reverse the order of the bit sequence to obtain the decimal number $(h_{ij}^F)_{10} = (b_{1:4}^F b_{5:8}^F)_{2}$. Performing this procedure for all pixels, we obtain the extracted secret image $H^G = \{h_{ij}^G\}$ as shown in Figure 1. In this method, the relationship between the stego and the extracted secret images, $H^F$ and $H^G$, is complementary to each other, i.e., when we view the stego image, the secret image is concealed in the stego image, on the other hand, when we view the extracted secret image, the stego image, which looks like the cover image, is concealed in the extracted secret image. Therefore, it is sufficient to save either $H^F$ or $H^G$, because the one of them can be produced from the other by reversing the order of every bit sequence. The algorithm for reversing the order of a bit sequence is summarized in Appendix A.

2.1.2 Error Diffusion The above bitwise operations embed 4 MSBs of a secret image in 4 LSBs of a cover image to produce a stego image. Therefore, the information in 4 LSBs of both cover and secret images is lost, that causes the deterioration of image quality in both cover and secret images. In this section, we propose a method for improving the image quality by using an error diffusion technique, which is a well-known technique in digital halftoning [15].

The proposed error diffusion method processes every pixel in a left-to-right raster scan order. Figure 2 illustrates a moment when the pixel $(i, j)$ colored in green is now being processed, where the shaded portion denotes that the pixels have already been processed, and white pixels denote unprocessed ones. In this situation, if the green pixel is on the border of an image, then a portion of the 3 × 3 mask illustrated in Figure 2 will go outside of the image region. In such cases, we would like to discard the errors going outside of the image region for computational simplicity and efficiency.

\[
\begin{align*}
\Delta e_{ij} & = f_{ij} - h_{ij}^F, \\
\Delta e_{ij} & = g_{ij} - h_{ij}^G,
\end{align*}
\]

(1) (2)

Assume that the pixel value $f_{ij}$ at the position $(i, j)$ in $F$ is changed into $h_{ij}^F$ by the above bitwise operations. Then we define the error between $f_{ij}$ and $h_{ij}^F$ by

$$
\Delta e_{ij} = f_{ij} - h_{ij}^F,
$$

which can be interpreted as an error between the cover and stego images. Similarly, we define the error between $g_{ij}$ and $h_{ij}^G$ by

$$
\Delta e_{ij} = g_{ij} - h_{ij}^G,
$$

which can be interpreted as an error between the original secret and extracted secret images. These errors in (1) and (2) are diffused into the unprocessed neighboring pixels in $F$ and $G$, respectively, where we would like to use the error diffusion coefficients presented by Floyd and Steinberg [16] as shown in Table 1, where '-' and '#' denote the processed and current pixels, respectively, and only unprocessed pixels have the positive values: $w_{0,1}$, $w_{1,-1}$, $w_{1,0}$ and $w_{1,1}$. The pixel values of unprocessed pixels are updated as follows:

$$f_{i+k,j+l} \leftarrow f_{i+k,j+l} + w_{k,l}e_{i,j}^f, \quad (3)$$
$$g_{i+k,j+l} \leftarrow g_{i+k,j+l} + w_{k,l}e_{i,j}^g, \quad (4)$$

where $k$ and $l$ are the indices in $w_{k,l}$ given by $(k, l) \in \mathcal{D}$, where $\mathcal{D}$ is a set of all pairs of two indices defined by $\mathcal{D} = \{(0,1), (1,-1), (1,0), (1,1)\}$, for indexing the unprocessed neighboring pixels at any position $(i, j)$ in $F$ or $G$, and $w_{k,l}$ denotes the error diffusion coefficients given in Table 1.

Table 1: Error diffusion coefficients by Floyd-Steinberg [16].

<table>
<thead>
<tr>
<th>$w_{-1,-1}$</th>
<th>$w_{-1,0}$</th>
<th>$w_{-1,1}$</th>
<th>$w_{0,1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
<td>$\frac{1}{16}$</td>
</tr>
</tbody>
</table>

After the update of the pixel values by (3) and (4), we proceed to the next pixel, where the bitwise operations are executed for the updated pixel values of $f_i$ and $g_i$. Such a procedure is applied to all pixels in $F$ and $G$ in the raster scan order. We summarize this error diffusion procedure for grayscale images in Algorithm 1, which is the proposed encoding algorithm, and the decoding algorithm is given by Algorithm 5 in Appendix A.

**Algorithm 1**

**Input:** two grayscale images $F = [f_{ij}]$ and $G = [g_{ij}]$ for cover and secret images

**Output:** a stego image $\tilde{H}^F = [\tilde{h}_{ij}^f]$  
1. for $i \leftarrow 1$ to $m$ do
2. for $j \leftarrow 1$ to $n$ do
3. Round $f_{ij}$ to 8-bit integer.
4. Round $g_{ij}$ to 8-bit integer.
5. (* bitwise operations *)
6. Convert $(f_{ij})_{10}$ into $(b_{f_{ij}}^G b_{f_{ij}}^F)_2$;
7. Convert $(g_{ij})_{10}$ into $(b_{g_{ij}}^G b_{g_{ij}}^F)_2$;
8. Extract 4 MSBs $b_{f_{ij}}^G$ and $b_{g_{ij}}^G$ from them;
9. Reverse the order of $b_{f_{ij}}^G$ as $b_{f_{ij}}^G$;
10. Combine $b_{f_{ij}}^F$ and $b_{g_{ij}}^G$ as $(b_{f_{ij}}^F b_{g_{ij}}^G)_4$;
11. Convert $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$ into $(b_{ij})_{10}$;
12. Reverse the order of $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$ as $(b_{f_{ij}}^G b_{f_{ij}}^F)_2$;
13. Convert $(b_{f_{ij}}^G b_{f_{ij}}^F)_2$ into $(\tilde{h}_{ij})_{10}$;
14. (* error diffusion *)
15. Compute the error between $f_{ij}$ and $\tilde{h}_{ij}$ by $e_{ij}^f = f_{ij} - \tilde{h}_{ij}$;
16. Compute the error between $g_{ij}$ and $\tilde{h}_{ij}$ by $e_{ij}^g = g_{ij} - \tilde{h}_{ij}$;
17. for $(k, l) \in \mathcal{D}$ do
18. if $(i + k, j + l) \in \Omega$ then
19. $f_{i+k,j+l} \leftarrow f_{i+k,j+l} + w_{k,l}e_{i,j}^f$;
20. $g_{i+k,j+l} \leftarrow g_{i+k,j+l} + w_{k,l}e_{i,j}^g$;
21. return $\tilde{H}^F = [\tilde{h}_{ij}^f]$;

In this algorithm, the output stego image $\tilde{H}^F = [\tilde{h}_{ij}^f]$ is distinguished from another output stego image $H^F = [h_{ij}^f]$, which is computed without error diffusion, by ‘‘~’’ (tilde).

The procedure for extracting the concealed secret image $H^G = [h_{ij}^G]$ from the stego image $\tilde{H}^F$ is the same as that for extracting $H^F$ from $H^G$, and is summarized as follows: For every pixel value $(\tilde{h}_{ij})_{10}$ of $\tilde{H}^F$, the order of the bit sequence $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$ converted from $(\tilde{h}_{ij})_{10}$ is reversed to obtain the corresponding pixel value $(h_{ij})_{10} = (b_{f_{ij}}^F b_{g_{ij}}^G)_2$ of $H^G$. This procedure is described in Algorithm 2.

**Algorithm 2**

**Input:** stego image $\tilde{H}^F = [\tilde{h}_{ij}^f]$

**Output:** concealed secret image $H^G = [h_{ij}^G]$

1. for $i \leftarrow 1$ to $m$ do
2. for $j \leftarrow 1$ to $n$ do
3. Convert $(\tilde{h}_{ij})_{10}$ into $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$;
4. Reverse the order of $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$ as $(b_{g_{ij}}^F b_{f_{ij}}^G)_2$ by Algorithm 5;
5. Convert $(b_{f_{ij}}^F b_{g_{ij}}^G)_2$ into $(\tilde{h}_{ij})_{10}$;
6. return $H^G = [h_{ij}^G]$;

The above simple operation for reversing the order of the bit sequence at each pixel can extract the concealed secret image successfully, because the proposed error diffusion procedure in Algorithm 1 takes into account both the stego and concealed secret images simultaneously and equally.

### 2.2 Color Image Steganography

The adaptation of the above grayscale image steganography method to color images is straightforward; R, G and B channels are processed in parallel by Algorithm 1. After that, the processed three channels are combined into a color image. However, such parallel processing requires three times of implementation of the error diffusion procedure for each color image. Alternatively, it may be useful that the procedure is described with vectors for array programming such as MATLAB and Python. In this section, we would like to describe the proposed color image steganography method with vectors in detail, where the error diffusion procedure is implemented only once.

Let $F = [f_{ij}]$ and $G = [g_{ij}]$ be two color images, where $f_{ij}$ and $g_{ij}$ denote the color pixel values at the position $(i, j)$ in $F$ and $G$, respectively, and have the following expressions: $f_{ij} = [f_{ij}^R, f_{ij}^G, f_{ij}^B]$ and $g_{ij} = [g_{ij}^R, g_{ij}^G, g_{ij}^B]$, where $f_{ij}^R = (g_{ij}^R), f_{ij}^G = (g_{ij}^G)$ and $f_{ij}^B = (g_{ij}^B)$ denote the red (R), green (G) and blue (B) values of the pixel in $F$ ($G$, respectively). The procedure of the proposed color image steganography is also divided into two parts: bitwise operations and error diffusion as follows.
2.2.1 Bitwise Operations

Assume that \( F \) and \( G \) are cover and secret images, respectively, and the elements of \( f_{ij} \) and \( g_{ij} \) are expressed in the decimal system, i.e., \( f_{ij} \in [0, 1, \ldots, 255] \) and \( g_{ij} \in [0, 1, \ldots, 255] \) for \( X \in [R, G, B] \) for 24-bit color images. We first convert each element \( (f_{ij})^{10} \) of \( f_{ij} \) into the corresponding binary number \( (b_{i,j}^R, b_{i,j}^G, b_{i,j}^B)_2 \), or \( (b_{i,j}^R, b_{i,j}^G, b_{i,j}^B)_2 \). When we view the extracted color secret image, the error distribution, by ‘\( \tilde{\cdot} \)’ (tilde).

\[ f_{i,j}^{10} \leftarrow f_{i,j} + w_{ik}e_{ij}^{F}, \quad (7) \]
\[ g_{i,j}^{10} \leftarrow g_{i,j} + w_{ik}e_{ij}^{G}, \quad (8) \]
where \( w_{ik} \) for \( (k, j) \in \{0, 1, (1, -1), (1, 0), (1, 1)\} \) denotes the error diffusion coefficients in Table 1.

After the update of the color pixel values by (7) and (8), we proceed to the next pixel, where the bitwise operations are executed for the updated color pixel values of \( f_{ij} \) and \( g_{ij} \). Such a procedure is applied to all color pixels in \( F \) and \( G \) in the raster scan order. We summarize this error diffusion procedure for color images in Algorithm 3.

Algorithm 3

1. for \( i \leftarrow 1 \) to \( m \) do
2. for \( j \leftarrow 1 \) to \( n \) do
3. for \( X \in [R, G, B] \) do
4. Round \( f_{ij} \) to 8-bit integer.
5. Round \( g_{ij} \) to 8-bit integer.
6. (* bitwise operations *)
7. Convert \( (f_{ij})^{10} \) into \( (b_{i,j}^R, b_{i,j}^G, b_{i,j}^B)_2 \).
8. Convert \( (g_{ij})^{10} \) into \( (b_{i,j}^R, b_{i,j}^G, b_{i,j}^B)_2 \).
9. Extract 4 MSBs \( b_{i,j}^R, b_{i,j}^G, b_{i,j}^B \) from them;
10. Reverse the order of \( b_{i,j}^R, b_{i,j}^G, b_{i,j}^B \) as \( b_{i,j}^{4:1} \).
11. Combine \( b_{i,j}^R, b_{i,j}^G, b_{i,j}^B \) as \( b_{i,j}^{4:1} \).
12. Convert \( b_{i,j}^{4:1} \) into \( (\tilde{b}_{i,j})^{10} \).
13. Reverse the order of \( b_{i,j}^{4:1} \) as \( b_{i,j}^{4:1} \).
14. Convert \( b_{i,j}^{4:1} \) into \( (\tilde{b}_{i,j})^{10} \).
15. \( h_{i,j}^{F} = [f_{ij}^{G}, f_{ij}^{G}, f_{ij}^{G}] \).
16. \( h_{i,j}^{G} = [g_{ij}^{f}, g_{ij}^{f}, g_{ij}^{f}] \).
17. (* error diffusion *)
18. Compute the error between \( f_{ij} \) and \( \tilde{h}_{ij}^{F} \) by \( e_{ij}^{F} = f_{ij} - \tilde{h}_{ij}^{F} \).
19. Compute the error between \( g_{ij} \) and \( \tilde{h}_{ij}^{G} \) by \( e_{ij}^{G} = g_{ij} - \tilde{h}_{ij}^{G} \).
20. for \( (k, l) \in \Omega \) do
21. if \( (i + k, j + l) \in \Omega \) then
22. \( f_{i+k,j+l} \leftarrow f_{i+k,j+l} + w_{ik}e_{ij}^{F} \).
23. \( g_{i+k,j+l} \leftarrow g_{i+k,j+l} + w_{ik}e_{ij}^{G} \).
24. return \( \tilde{H}^{F} = [\tilde{h}_{ij}^{F}] \).

In this algorithm, the output stego image \( \tilde{H}^{F} = [\tilde{h}_{ij}^{F}] \) is distinguished from another output stego image \( \tilde{H}^{F} = [\tilde{h}_{ij}^{F}] \), which is computed without error diffusion, by ‘\( \tilde{\cdot} \)’ (tilde).

The procedure for extracting the concealed secret image \( \tilde{H}^{G} \) from the stego image \( \tilde{H}^{F} \) produced by Algorithm 3 is...
the same as that for extracting $H^G$ from $H^F$, and is summarized as follows: For every color pixel value $(\tilde{h}_{ij}^{FX})_{10}$ of $\tilde{H}^F$ for $X \in \{R, G, B\}$, the order of the bit sequence $(b_{FX}^{1,4}, b_{FX}^{2,4})$ converted from $(\tilde{h}_{ij}^{FX})_{10}$ is reversed to obtain the corresponding color pixel value $(\tilde{h}_{ij}^{GX})_{10} = (b_{GX}^{1,4}, b_{GX}^{2,4})_{10}$ of $\tilde{H}^G$ for $X \in \{R, G, B\}$. This procedure is described in Algorithm 4.

**Algorithm 4**

**Input:** stego image $\tilde{H}^F = [\tilde{h}_{ij}^F]$ where $\tilde{h}_{ij}^F = [\tilde{h}_{ij}^{FR}, \tilde{h}_{ij}^{FG}, \tilde{h}_{ij}^{FB}]$

**Output:** concealed secret image $\tilde{H}^G = [\tilde{h}_{ij}^G]$ where $\tilde{h}_{ij}^G = [\tilde{h}_{ij}^{GR}, \tilde{h}_{ij}^{GG}, \tilde{h}_{ij}^{GB}]$

1. for $i \leftarrow 1$ to $m$ do
2. for $j \leftarrow 1$ to $n$ do
3. for $X \in \{R, G, B\}$ do
4. Convert $(\tilde{h}_{ij}^{FX})_{10}$ into $(b_{FX}^{1,4}, b_{FX}^{2,4})$;
5. Reverse the order of $(b_{FX}^{1,4}, b_{FX}^{2,4})$ as $(b_{GX}^{1,4}, b_{GX}^{2,4})_2$ by Algorithm 5;
6. Convert $(b_{GX}^{1,4}, b_{GX}^{2,4})_2$ into $(\tilde{h}_{ij}^{GX})_{10}$;
7. return $\tilde{H}^G = [\tilde{h}_{ij}^G]$ where $\tilde{h}_{ij}^G = [\tilde{h}_{ij}^{GR}, \tilde{h}_{ij}^{GG}, \tilde{h}_{ij}^{GB}]$

As well as Algorithm 2, the above simple operation for reversing the order of the bit sequence at each pixel and color channel can also extract the concealed secret color image successfully, because the proposed error diffusion procedure in Algorithm 3 takes into account both the stego and concealed secret color images simultaneously and equally.

3. Experimental Results

In this section, we show experimental results on the standard image database, SIDBA [17] (it can be downloaded from http://www.eess.ics.kanagawa-it.ac.jp/app_images_j.html), in which the number of pixels in each image is $256 \times 256$ for both grayscale and color images. We first show the results of grayscale image steganography, and then show the results of color image steganography.

### 3.1 Grayscale Image Steganography

Figure 3 shows an example of the proposed two-in-one grayscale image steganography, where Figures 3(a) and (d) show the input cover and secret images $F$ and $G$, which are combined into the stego image $\tilde{H}^F$ in Figure 3(b) without error diffusion. From the stego image $\tilde{H}^F$ in Figure 3(b), we can extract the concealed secret image $\tilde{H}^G$ as shown in Figure 3(e) by reversing the order of bit sequences. On the other hand, if we feed the input images $F$ and $G$ into Algorithm 1, then we obtain the stego image $\tilde{H}^F$ in Figure 3(c), from which we can extract the concealed secret image $\tilde{H}^G$ in Figure 3(f) by Algorithm 2. The stego and the extracted secret images $\tilde{H}^F$ and $\tilde{H}^G$ produced without error diffusion procedure in Figures 3(b) and (e) have conspicuous noise like false contours [18]. On the other hand, using error diffusion procedure, we have visually preferable results in Figures 3(c) and (f), where the noises are suppressed well.

In order to show this effect of the error diffusion procedure more clearly, we zoomed the parts of Figures 3(a), (b) and (c) in Figures 4(a), (b) and (c), respectively, where the false contours visible on the face and shoulder of woman in Figure 4(b) are suppressed well in Figure 4(c) which is similar to the original cover image in Figure 4(a).

### 3.2 Color Image Steganography

Figure 5 shows an example of the proposed two-in-one color image steganography, where Figures 5(a) and (d) show the input cover and secret images, which are combined into the stego image $\tilde{H}^F$ in Figure 5(b) without error diffusion. From the stego image $\tilde{H}^F$ in Figure 5(b), we can extract the concealed secret image as shown in Figure 5(e) by reversing the order of bit sequences. On the other hand, if we feed the input images in Figures 5(a) and (d) into Algorithm 3, then we obtain the stego image $\tilde{H}^F$ in Figure 5(c), from which we can extract the concealed secret image $\tilde{H}^G$ in Figure 5(f) by Algorithm 4. The stego and the extracted secret images $\tilde{H}^F$ and $\tilde{H}^G$ produced without error diffusion procedure in Figures 5(b) and (e) have conspicuous noise like false contours [18] as well as the above results for grayscale images. On the other hand, using error diffusion procedure, we have visually preferable results in Figures 5(c) and (f), where the noises are suppressed well.

In order to show this effect of the error diffusion proce-
The limitation of the proposed method is the robustness to lossy image compression such as JPEG, which will mainly distort the secret image embedded in the stego image by the proposed method. To alleviate this kind of distortion, we are planning to improve the proposed method to more robust one by reordering the bit sequence. The proposed error diffusion method can also be used in the improved version of the proposed method.
Figure 7: Six pairs of cover and secret images in the left part of this figure are used for quantitative evaluation with SSIM [19]. MSE and SNR. The middle part of this figure shows the stego and extracted secret images given by the proposed algorithm without error diffusion, and the corresponding SSIM, MSE and SNR values are shown on the right side of the images. The right part of this figure shows the stego and extracted secret images given by Algorithms 1 and 3, both of which use the error diffusion method, and the corresponding SSIM, MSE and SNR values are shown on the right sides of the images. All evaluated values in the right part are improved compared with that in the middle part.

5. Conclusions

In this paper, we proposed an image steganography method for concealing a secret image into a cover image of the same size. The proposed method is based on a least significant bit (LSB) approach, where the most significant bits (MSBs) of the secret image are embedded in the LSBs of the cover image after the reversal of the order of the bit sequences. The concealed secret image can be extracted by reversing the order of every bit sequence converted from each pixel value of the stego image. That is, the reversal of the order of bit sequences switches the role of LSBs and MSBs in the proposed image steganography method. Furthermore, we proposed a method for improving the image quality of both stego and concealed secret images by using an error diffusion method. Experimental results revealed that the proposed method can be used for both grayscale and color image steganography, and the proposed error diffusion method improves the image quality of both stego and extracted secret images visually and quantitatively.

Appendix

A. Reversing the Order of a Bit Sequence

Let \((b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8)\) be a bit sequence. Then we can reverse the order of the bit sequence as follows: First, we compute two bitwise ANDs: \((b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8) \land (0101 0101)_2 = (0b_2 0b_1 0b_0 0b_9)_2 = x\) and \((b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8) \land (1010 1010)_2 = (b_0 b_1 0b_2 0b_3)_2 = y\), where \(\land\) denotes the logical AND operator. Next, we compute \((x < 1) \lor (y \geq 1) = (b_0 b_3 0b_2 0b_0 0b_1)_2 \lor (0b_0 b_1 0b_0 b_2)_2 = (b_0 b_1 b_0 b_3 b_2 b_3 b_0 b_2)_2 = z\), where \(\lor\) denotes the logical OR operator, and \(<\) and \(\geq\) denote the left and right logical shift operators, respectively. Then, we compute the following bitwise ANDs: \(z \land (0011 0011)_2 = (00b_2 b_1 00b_0 b_7)_2 = u\) and \(z \land (1101 1101)_2 = (b_2 b_0 00 b_1 00 b_0 b_9)_2 = v\). After that, we compute \((u \ll 2) \lor (v \gg 2) = (b_2 b_1 00 b_0 b_7)_2 \lor (00b_2 b_1 00b_0 b_3)_2 = (b_2 b_2 b_0 b_3 b_0 b_1 b_7 b_0)_2 = w\). Finally, we compute \((w \ll 4) \lor (w \gg 4) = (b_2 b_2 b_0 b_3 b_0 0000)_2 \lor (0000 b_2 b_2 b_1)_2 = (b_2 b_0 b_2 b_0 b_3 b_2 b_2 b_1)_2\), which is the order-reversed bit sequence of the original one. This procedure is summarized in Algorithm 5.

Algorithm 5

Input: a bit sequence \((b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8)_2\)

Output: an order-reversed bit sequence \((b_2 b_0 b_2 b_3 b_0 b_3 b_2 b_1)_2\)

1. Compute \(x = (b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8) \land (0101 0101)_2\);
2. Compute \(y = (b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8) \land (1010 1010)_2\);
3. Compute \(z = (x < 1) \lor (y \geq 1)\);
4. Compute \(u = z \land (0011 0011)_2\);
5. Compute \(v = z \land (1101 1101)_2\);
6. Compute \(w = (u \ll 2) \lor (v \gg 2)\);
7. \(\text{return} (w \ll 4) \lor (w \gg 4)\).
Figure 8: Six pairs of cover and secret images in the left part of this figure are selected from the Image Processing Toolbox for MATLAB, used for quantitative evaluation with SSIM [19], MSE and SNR. The middle part of this figure shows the stego and extracted secret images given by the proposed algorithm without error diffusion, and the corresponding SSIM, MSE and SNR values are shown on the right side of the images. The right part of this figure shows the stego and extracted secret images given by Algorithms 1 and 3, both of which use the error diffusion method, and the corresponding SSIM, MSE and SNR values are shown on the right sides of the images. All evaluated values in the right part are improved compared with that in the middle part.

Acknowledgment
This work was supported by JSPS KAKENHI Grant Number JP16H03019.

References


[8] B. Li, J. He, J. Huang, Y. Q. Shi, “A Survey on Image


Ruixi Dong (Non-member) She received B.A. degree from Dalian Jiaotong University of China in 2017. She is currently a graduate student in Kyushu University. Her research interests include image steganography and image processing.

Kohei Inoue (Member) He received B.Des., M.Des. and D.Eng. degrees from Kyushu Institute of Design in 1996, 1998 and 2000, respectively. He is currently an Associate Professor in Kyushu University. His research interests include pattern recognition and image processing.

Kenji Hara (Non-member) He received the BE and ME degrees from Kyoto University in 1987 and 1989, respectively, and the PhD degree from Kyushu University in 1999. He is currently an Associate Professor in Kyushu University. His research interests include physics-based vision and geometric modeling.

Kiichi Urahama (Member) He received M.Eng. and D.Eng. degrees from Kyushu University in 1976 and 1980. From 1980 to 1995 he was an Associate Professor in Kyushu Institute of Technology. He is now a Professor in Kyushu University. His research interests include pattern recognition, image processing and computer graphics.