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

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：

Journal of the Institute of Industrial Applications Engineers Vol.7, No.2, pp.42–50, (2019.4.25)
DOI: 10.12792/JIIAE.7.42 Online edition: ISSN 2187-8811 Print edition: ISSN 2188-1758

Paper

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 attentions 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 $(b_1^F b_2^F b_3^F b_4^F \ b_5^F b_6^F b_7^F b_8^F)_2$, 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: $b_{1:4}^F = b_1^F b_2^F b_3^F b_4^F$. 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 $b_{4:1}^G = b_4^G b_3^G b_2^G b_1^G$, and combine it with $b_{1:4}^F$ as $(b_{1:4}^F \ b_{4:1}^G)_2$ which is converted into the decimal number as $(h_{ij}^F)_{10} = (b_{1:4}^F \ b_{4:1}^G)_2$ 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.

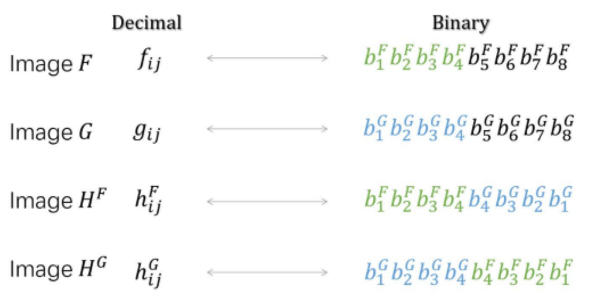


Figure 1: Conversion between decimal and binary numbers of pixel values.

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 $(b_{1:4}^F \ b_{4:1}^G)_2$, and then reverse the order of the bit sequence to obtain the decimal number $(h_{ij}^G)_{10} = (b_{1:4}^G \ b_{4:1}^F)_2$. 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.

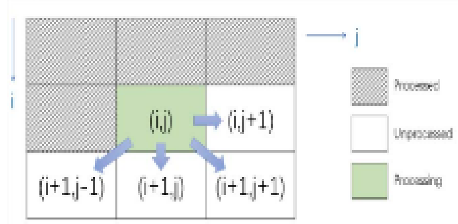


Figure 2: Error diffusion from a pixel (i, j) to the neighboring pixels.

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

$$e_{ij}^F = f_{ij} - h_{ij}^F, \tag{1}$$

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

$$e_{ij}^G = g_{ij} - h_{ij}^G, \tag{2}$$

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:

$$f_{i+k,j+l} \leftarrow f_{i+k,j+l} + w_{k,l}e^F_{ij}, \tag{3}$$

$$g_{i+k,j+l} \leftarrow g_{i+k,j+l} + w_{k,l}e^G_{ij}, \tag{4}$$

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 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].

-	-	-
-	#	$w_{0,1} = \frac{7}{16}$
$w_{1,-1} = \frac{3}{16}$	$w_{1,0} = \frac{5}{16}$	$w_{1,1} = \frac{1}{16}$

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_{ij} and g_{ij}. 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 $\tilde{H}^F = [\tilde{h}^F_{ij}]$

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 $(b^F_{1:4}\ b^F_{5:8})_2$;
7. Convert $(g_{ij})_{10}$ into $(b^G_{1:4}\ b^G_{5:8})_2$;
8. Extract 4 MSBs $b^F_{1:4}$ and $b^G_{1:4}$ from them;
9. Reverse the order of $b^G_{1:4}$ as $b^G_{4:1}$;
10. Combine $b^F_{1:4}$ and $b^G_{4:1}$ as $(b^F_{1:4}\ b^G_{4:1})_2$;
11. Convert $(b^F_{1:4}\ b^G_{4:1})_2$ into $(\tilde{h}^F_{ij})_{10}$;
12. Reverse the order of $(b^F_{1:4}\ \ b^G_{4:1})_2$ as $(b^G_{1:4}\ b^F_{4:1})_2$;
13. Convert $(b^G_{1:4}\ b^F_{4:1})_2$ into $(\tilde{h}^G_{ij})_{10}$;
14. ($*$ error diffusion $*$)
15. Compute the error between f_{ij} and h^F_{ij} by $e^F_{ij} = f_{ij} - \tilde{h}^F_{ij}$;
16. Compute the error between g_{ij} and h^G_{ij} by $e^G_{ij} = g_{ij} - \tilde{h}^G_{ij}$;
17. **for** $(k,l) \in \mathcal{D}$ **do**
18. **if** $(i+k, j+l) \in \Omega$ **then**
19. $f_{i+k,j+l} \leftarrow f_{i+k,j+l} + w_{k,l}e^F_{ij}$;
20. $g_{i+k,j+l} \leftarrow g_{i+k,j+l} + w_{k,l}e^G_{ij}$;
21. **return** $\tilde{H}^F = [\tilde{h}^F_{ij}]$

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

The procedure for extracting the concealed secret image $\tilde{H}^G = [\tilde{h}^G_{ij}]$ from the stego image $\tilde{H}^F$ is the same as that for extracting H^G from H^F, and is summarized as follows: For every pixel value $(\tilde{h}^F_{ij})_{10}$ of $\tilde{H}^F$, the order of the bit sequence $(b^F_{1:4}\ b^G_{4:1})_2$ converted from $(\tilde{h}^F_{ij})_{10}$ is reversed to obtain the corresponding pixel value $(\tilde{h}^G_{ij})_{10} = (b^G_{1:4}\ b^F_{4:1})_2$ of $\tilde{H}^G$. This procedure is described in Algorithm 2.

Algorithm 2

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

Output: concealed secret image $\tilde{H}^G = [\tilde{h}^G_{ij}]$

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

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 $\boldsymbol{F} = [\boldsymbol{f}_{ij}]$ and $\boldsymbol{G} = [\boldsymbol{g}_{ij}]$ be two color images, where $\boldsymbol{f}_{ij}$ and $\boldsymbol{g}_{ij}$ denote the color pixel values at the position (i, j) in $\boldsymbol{F}$ and $\boldsymbol{G}$, respectively, and have the following expressions: $\boldsymbol{f}_{ij} = [f^R_{ij}, f^G_{ij}, f^B_{ij}]$ and $\boldsymbol{g}_{ij} = [g^R_{ij}, g^G_{ij}, g^B_{ij}]$, where f^R_{ij} (g^R_{ij}), f^G_{ij} (g^G_{ij}) and f^B_{ij} (g^B_{ij}) denote the red (R), green (G) and blue (B) values of the pixel in $\boldsymbol{F}$ $(\boldsymbol{G})$, 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 $\boldsymbol{F}$ and $\boldsymbol{G}$ are cover and secret images, respectively, and the elements of $\boldsymbol{f}_{ij}$ and $\boldsymbol{g}_{ij}$ are expressed in the decimal system, i.e., $f_{ij}^X \in \{0, 1, \ldots, 255\}$ and $g_{ij}^X \in \{0, 1, \ldots, 255\}$ for $X \in \{R, G, B\}$ for 24-bit color images. We first convert each element $(f_{ij}^X)_{10}$ of $\boldsymbol{f}_{ij}$ into the corresponding binary number $(b_1^{FX} b_2^{FX} b_3^{FX} b_4^{FX}\ b_5^{FX} b_6^{FX} b_7^{FX} b_8^{FX})_2$ or $(b_{1:4}^{FX}\ b_{5:8}^{FX})_2$ for $X \in \{R, G, B\}$, where $b_k^{FX} \in \{0, 1\}$ for $k = 1, 2, \ldots, 8$. Similarly, we also convert $(g_{ij}^X)_{10}$ into the binary number $(b_{1:4}^{GX}\ b_{5:8}^{GX})_2$ for $X \in \{R, G, B\}$. Next, we extract 4 MSBs from those bit sequences as $b_{1:4}^{FX}$ and $b_{1:4}^{GX}$. Then we reverse the order of the latter $b_{1:4}^{GX}$ as $b_{4:1}^{GX}$, and combine it with $b_{1:4}^{FX}$ as $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ which is converted into the decimal number as $(h_{ij}^{FX})_{10} = (b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ for $X \in \{R, G, B\}$. This procedure is performed for all pixels. As a results, we obtain the color stego image $\boldsymbol{H}^F = [\boldsymbol{h}_{ij}^F]$, where $\boldsymbol{h}_{ij}^F = [h_{ij}^{FR}, h_{ij}^{FG}, h_{ij}^{FB}]$, which appears to be similar to $\boldsymbol{F}$, but conceals a partial information of $\boldsymbol{G}$ in it.

The procedure for extracting the concealed secret image from the stego image $\boldsymbol{H}^F$ is as follows: For each $X \in \{R, G, B\}$, we convert the decimal pixel value $(h_{ij}^{FX})_{10}$ of $\boldsymbol{H}^F$ into the binary number $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$, and then reverse the order of the bit sequence to obtain the decimal number $(h_{ij}^{GX})_{10} = (b_{1:4}^{GX}\ b_{4:1}^{FX})_2$. Performing this procedure for all pixels, we obtain the extracted secret image $\boldsymbol{H}^G = [\boldsymbol{h}_{ij}^G]$ where $\boldsymbol{h}_{ij}^G = [h_{ij}^{GR}, h_{ij}^{GG}, h_{ij}^{GB}]$. In this method, the relationship between the color stego and the extracted color secret images, $\boldsymbol{H}^F$ and $\boldsymbol{H}^G$, is also complementary to each other, i.e., when we view the color stego image, the color secret image is concealed in the color stego image, on the other hand, when we view the extracted color secret image, the color stego image, which looks like the color cover image, is concealed in the extracted color secret image. Therefore, it is enough to save either $\boldsymbol{H}^F$ or $\boldsymbol{H}^G$, because the one of them can be produced from the other by reversing the order of every bit sequence.

2.2.2 Error Diffusion We also improve the image quality of the above color stego image $\boldsymbol{H}^F = [\boldsymbol{h}_{ij}^F]$ and extracted secret image $\boldsymbol{H}^G = [\boldsymbol{h}_{ij}^G]$ by using an error diffusion method as well as the grayscale version.

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

$$\boldsymbol{e}_{ij}^F = \boldsymbol{f}_{ij} - \boldsymbol{h}_{ij}^F, \tag{5}$$

which can be interpreted as an error between the color cover and stego images. Similarly, we define the error between $\boldsymbol{g}_{ij}$ and $\boldsymbol{h}_{ij}^G$ by

$$\boldsymbol{e}_{ij}^G = \boldsymbol{g}_{ij} - \boldsymbol{h}_{ij}^G, \tag{6}$$

which can be interpreted as an error between the original color secret and extracted secret images. These errors in (5) and (6) are diffused into the unprocessed neighboring pixels as well as the above error diffusion procedure for grayscale images as follows:

$$\boldsymbol{f}_{i+k,j+l} \leftarrow \boldsymbol{f}_{i+k,j+l} + w_{k,l}\boldsymbol{e}_{ij}^F, \tag{7}$$

$$\boldsymbol{g}_{i+k,j+l} \leftarrow \boldsymbol{g}_{i+k,j+l} + w_{k,l}\boldsymbol{e}_{ij}^G, \tag{8}$$

where $w_{k,l}$ 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 $\boldsymbol{f}_{ij}$ and $\boldsymbol{g}_{ij}$. Such a procedure is applied to all color pixels in $\boldsymbol{F}$ and $\boldsymbol{G}$ in the raster scan order. We summarize this error diffusion procedure for color images in Algorithm 3.

Algorithm 3

Input: two color images $\boldsymbol{F} = [\boldsymbol{f}_{ij}]$ and $\boldsymbol{G} = [\boldsymbol{g}_{ij}]$ for cover and secret images

Output: a stego image $\tilde{\boldsymbol{H}}^F = [\tilde{\boldsymbol{h}}_{ij}^F]$

for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** n **do**
    for $X \in \{R, G, B\}$ **do**
      Round f_{ij}^X to 8-bit integer.
      Round g_{ij}^X to 8-bit integer.
      (∗ bitwise operations ∗)
      Convert $(f_{ij}^X)_{10}$ into $(b_{1:4}^{FX}\ b_{5:8}^{FX})_2$;
      Convert $(g_{ij}^X)_{10}$ into $(b_{1:4}^{GX}\ b_{5:8}^{GX})_2$;
      Extract 4 MSBs $b_{1:4}^{FX}$ and $b_{1:4}^{GX}$ from them;
      Reverse the order of $b_{1:4}^{GX}$ as $b_{4:1}^{GX}$;
      Combine $b_{1:4}^{FX}$ and $b_{4:1}^{GX}$ as $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$;
      Convert $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ into $(\tilde{h}_{ij}^{FX})_{10}$;
      Reverse the order of $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ as $(b_{1:4}^{GX}\ b_{4:1}^{FX})_2$;
      Convert $(b_{1:4}^{GX}\ b_{4:1}^{FX})_2$ into $(\tilde{h}_{ij}^{GX})_{10}$;
    $\tilde{\boldsymbol{h}}_{ij}^F = [\tilde{h}_{ij}^{FR}, \tilde{h}_{ij}^{FG}, \tilde{h}_{ij}^{FB}]$;
    $\tilde{\boldsymbol{h}}_{ij}^G = [\tilde{h}_{ij}^{GR}, \tilde{h}_{ij}^{GG}, \tilde{h}_{ij}^{GB}]$;
    (∗ error diffusion ∗)
    Compute the error between $\boldsymbol{f}_{ij}$ and $\tilde{\boldsymbol{h}}_{ij}^F$ by $\boldsymbol{e}_{ij}^F = \boldsymbol{f}_{ij} - \tilde{\boldsymbol{h}}_{ij}^F$;
    Compute the error between $\boldsymbol{g}_{ij}$ and $\tilde{\boldsymbol{h}}_{ij}^G$ by $\boldsymbol{e}_{ij}^G = \boldsymbol{g}_{ij} - \tilde{\boldsymbol{h}}_{ij}^G$;
    for $(k, l) \in \mathcal{D}$ **do**
      if $(i + k, j + l) \in \Omega$ **then**
        $\boldsymbol{f}_{i+k,j+l} \leftarrow \boldsymbol{f}_{i+k,j+l} + w_{k,l}\boldsymbol{e}_{ij}^F$;
        $\boldsymbol{g}_{i+k,j+l} \leftarrow \boldsymbol{g}_{i+k,j+l} + w_{k,l}\boldsymbol{e}_{ij}^G$;
return $\tilde{\boldsymbol{H}}^F = [\tilde{\boldsymbol{h}}_{ij}^F]$

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

The procedure for extracting the concealed secret image $\tilde{\boldsymbol{H}}^G$ from the stego image $\tilde{\boldsymbol{H}}^F$ produced by Algorithm 3 is the same as that for extracting $\boldsymbol{H}^G$ from $\boldsymbol{H}^F$, and is summarized as follows: For every color pixel value $(\tilde{h}_{ij}^{FX})_{10}$ of $\tilde{\boldsymbol{H}}^F$ for $X \in \{R, G, B\}$, the order of the bit sequence $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ converted from $(\tilde{h}_{ij}^{FX})_{10}$ is reversed to obtain the corresponding color pixel value $(\tilde{h}_{ij}^{GX})_{10} = (b_{1:4}^{GX}\ b_{4:1}^{FX})_2$ of $\tilde{\boldsymbol{H}}^G$ for $X \in \{R, G, B\}$. This procedure is described in Algorithm 4.

Algorithm 4

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

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

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 $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$;
5. Reverse the order of $(b_{1:4}^{FX}\ b_{4:1}^{GX})_2$ as $(b_{1:4}^{GX}\ b_{4:1}^{FX})_2$ by Algorithm 5;
6. Convert $(b_{1:4}^{GX}\ b_{4:1}^{FX})_2$ into $(\tilde{h}_{ij}^{GX})_{10}$;
7. **return** $\tilde{\boldsymbol{H}}^G = [\tilde{\boldsymbol{h}}_{ij}^G]$ where $\tilde{\boldsymbol{h}}_{ij}^G = [\tilde{h}_{ij}^{GR}, \tilde{h}_{ij}^{GG}, \tilde{h}_{ij}^{GB}]$

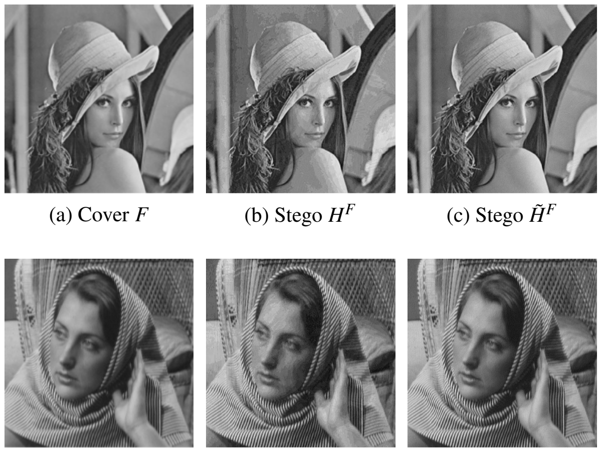
As well as Algorithm 2, the above simple operation for reversing the order of the bit sequence at each pixel and color channel can also extract the concealed secret color image successfully, because the proposed error diffusion procedure in Algorithm 3 takes into account both the stego and concealed 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.ess.ic.kanagawa-it.ac.jp/app_images_j.html`), in which the number of pixels in each image is 256×256 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 H^F in Figure 3(b) without error diffusion. From the stego image H^F in Figure 3(b), we can extract the concealed secret image 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 H^F and 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).



(a) Cover F (b) Stego H^F (c) Stego $\tilde{H}^F$

(d) Secret G (e) Extracted H^G (f) Extracted $\tilde{H}^G$

Figure 3: Example of two-in-one grayscale image steganography: (a) Cover image F. (b) Stego image H^F produced without error diffusion. (c) Stego image $\tilde{H}^F$ produced with error diffusion. (d) Secret image G. (e) Extracted secret image H^G from H^F. (f) Extracted secret image $\tilde{H}^G$ from $\tilde{H}^F$.

(a) Cover F (b) Stego H^F (c) Stego $\tilde{H}^F$

Figure 4: Zoomed parts of stego images: (a) F in Figure 3(a), (b) H^F in Figure 3(b), and (c) $\tilde{H}^F$ in Figure 3(c).

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 $\boldsymbol{H}^F$ in Figure 5(b) without error diffusion. From the stego image $\boldsymbol{H}^F$ 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 $\tilde{\boldsymbol{H}}^F$ in Figure 5(c), from which we can extract the concealed secret image $\tilde{\boldsymbol{H}}^G$ in Figure 5(f) by Algorithm 4. The stego and the extracted secret images $\boldsymbol{H}^F$ and $\boldsymbol{H}^G$ 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-

(a) Cover $\boldsymbol{F}$ (b) Stego $\boldsymbol{H}^F$ (c) Stego $\tilde{\boldsymbol{H}}^F$

(d) Secret $\boldsymbol{G}$ (e) Extracted $\boldsymbol{H}^G$ (f) Extracted $\tilde{\boldsymbol{H}}^G$

Figure 5: Example of two-in-one color image steganography: (a) Cover image $\boldsymbol{F}$. (b) Stego image $\boldsymbol{H}^F$ produced without error diffusion. (c) Stego image $\tilde{\boldsymbol{H}}^F$ produced with error diffusion. (d) Secret image $\boldsymbol{G}$. (e) Extracted secret image $\boldsymbol{H}^G$ from $\boldsymbol{H}^F$. (f) Extracted secret image $\tilde{\boldsymbol{H}}^G$ from $\tilde{\boldsymbol{H}}^F$.

dure more clearly, we zoomed the parts of Figures 5(d), (e) and (f) in Figures 6(a), (b) and (c), respectively, where the false contours visible on the face and balloon in Figure 6(b) are suppressed well in Figure 6(c) which is similar to the original secret image in Figure 6(a).

(a) Secret $\boldsymbol{G}$ (b) Extracted $\boldsymbol{H}^G$ (c) Extracted $\tilde{\boldsymbol{H}}^G$

Figure 6: Zoomed parts of extracted secret images: (a) $\boldsymbol{G}$ in Figure 5(d), (b) $\tilde{\boldsymbol{H}}^G$ in Figure 5(e), and (c) $\tilde{\boldsymbol{H}}^G$ in Figure 5(f).

Next, we show the results of quantitative evaluation of image quality using the structural similarity (SSIM) index [19], the mean squared error (MSE) and the signal-to-noise ratio (SNR) in Figure 7 including the results in Figures 3 and 5, in the left part of which, six pairs of the original color and grayscale images are selected from the SIDBA image database [17] as cover and secret images. The resultant stego and extracted images by the proposed algorithm without error diffusion and that with error diffusion are shown in the middle and right parts of Figure 7, respectively, with the corresponding SSIM, MSE and SNR values shown on the right sides of the images. For computing those values in Figure 7, we first smoothed the input and output images by Gaussian filter with the standard deviation 1.5, and then computed the SSIM, MSE and SNR values between the smoothed images. The smoothing by Gaussian filer is necessary for valid evaluation of image quality, which agrees with the evaluation based on human visual system. For all pairs in Figure 7, the evaluated values are improved (SSIM and SNR are increased, and MSE is decreased) by the proposed error diffusion method, which demonstrates the effectiveness of the proposed method.

Next, we show the results with other image dataset selected from the Image Processing Toolbox for MATLAB in Figure 8, where six pairs of the original color and grayscale images are shown in the left part, and the middle and right parts show the corresponding stego and extracted secret images with their quantitative evaluation results. As well as the results in Figure 7, we also observed in Figure 8 that the proposed error diffusion method improved the image quality for both the stego and extracted secret images.

4. Discussion

In the above experimental results, we have demonstrated the function of the proposed image steganography method for grayscale and color images. As well as Hadidi and Ibrahim's method [13], we used 4 LSBs for embedding secret information, which is an image of the same size as a cover image in our setting. That is, the proposed method maximally utilizes 4 LSBs of a cover image for embedding 4 MSBs of a secret image, which brings about a complementary between the stego and extracted secret images. In other words, two images, cover and secret images, are combined into one image, stego or extracted secret image, in which the one is given by reversing the order of the bit sequences of the other. Therefore, it is sufficient for users to save either the stego or the extracted secret image. In our future work, we would like to compare the proposed method with other image steganography methods which hide a full-size image into a carrier image as well as Baluja's deep steganography [14].

Furthermore, we improved the image quality of the stego and extracted secret images by using an error diffusion method. For a typical instance of existing error diffusion methods, we used the error diffusion coefficients proposed by Floyd and Steinberg [16] in Table 1. However, the other error diffusion coefficients may suitable for the proposed two-in-one image steganography. Our future research will include the optimization of the error diffusion coefficients for the proposed two-in-one image steganography. Moreover, the steganalysis for the proposed steganography method will also be our future work, where structural steganalysis methods [20] including RS [21] and WS [22] may be applicable.

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.

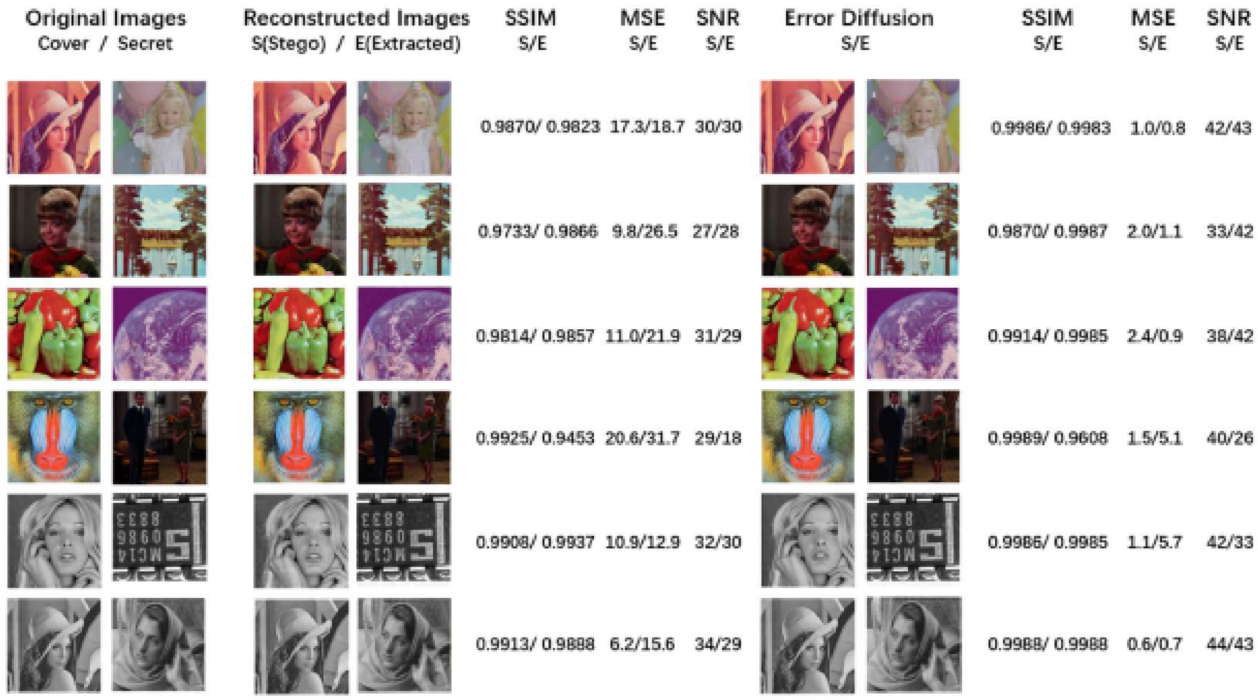


Figure 7: Six pairs of cover and secret images in the left part of this figure are used for quantitative evaluation with SSIM [19], MSE and SNR. The middle part of this figure shows the stego and extracted secret images given by the proposed algorithm without error diffusion, and the corresponding SSIM, 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 $(b_1b_2b_3b_4\ \ b_5b_6b_7b_8)_2$ be a bit sequence. Then we can reverse the order of the bit sequence as follows: First, we compute two bitwise ANDs: $(b_1b_2b_3b_4\ \ b_5b_6b_7b_8)_2 \wedge (0101\ 0101)_2 = (0b_20b_4\ 0b_60b_8)_2 = x$ and $(b_1b_2b_3b_4\ b_5b_6b_7b_8)_2 \wedge (1010\ 1010)_2 = (b_10b_30\ b_50b_70)_2 = y$, where $\wedge$ denotes the logical AND operator. Next, we compute $(x \ll 1) \vee (y \gg 1) = (b_20b_40\ b_60b_80)_2 \vee (0b_10b_3\ 0b_50b_7)_2 = (b_2b_1b_4b_3\ b_6b_5b_8b_7)_2 = z$, where $\vee$ denotes the logical OR operator, and $\ll$ and $\gg$ denote the left and right logical shift operators, respectively. Then, we compute the following bitwise ANDs: $z \wedge (0011\ 0011)_2 = (00b_4b_3\ 00b_8b_7)_2 = u$ and $z \wedge (1100\ 1100)_2 = (b_2b_100\ b_6b_500)_2 = v$. After that, we compute $(u \ll 2) \vee (v \gg 2) = (b_4b_300\ b_8b_700)_2 \vee (00b_2b_1\ 00b_6b_5)_2 = (b_4b_3b_2b_1\ b_8b_7b_6b_5)_2 = w$. Finally, we compute $(w \ll 4) \vee (w \gg 4) = (b_8b_7b_6b_5\ 0000)_2 \vee (0000\ b_4b_3b_2b_1)_2 = (b_8b_7b_6b_5\ b_4b_3b_2b_1)_2$, which is the order-reversed bit sequence of the original one. This procedure is summarized in Algorithm 5.

Algorithm 5

Input: a bit sequence $(b_1b_2b_3b_4\ b_5b_6b_7b_8)_2$

Output: an order-reversed bit sequence $(b_8b_7b_6b_5\ b_4b_3b_2b_1)_2$

1. Compute $x = (b_1b_2b_3b_4\ b_5b_6b_7b_8)_2 \wedge (0101\ 0101)_2$;
2. Compute $y = (b_1b_2b_3b_4\ b_5b_6b_7b_8)_2 \wedge (1010\ 1010)_2$;
3. Compute $z = (x \ll 1) \vee (y \gg 1)$;
4. Compute $u = z \wedge (0011\ 0011)_2$;
5. Compute $v = z \wedge (1100\ 1100)_2$;
6. Compute $w = (u \ll 2) \vee (v \gg 2)$;
7. **return** $(w \ll 4) \vee (w \gg 4)$

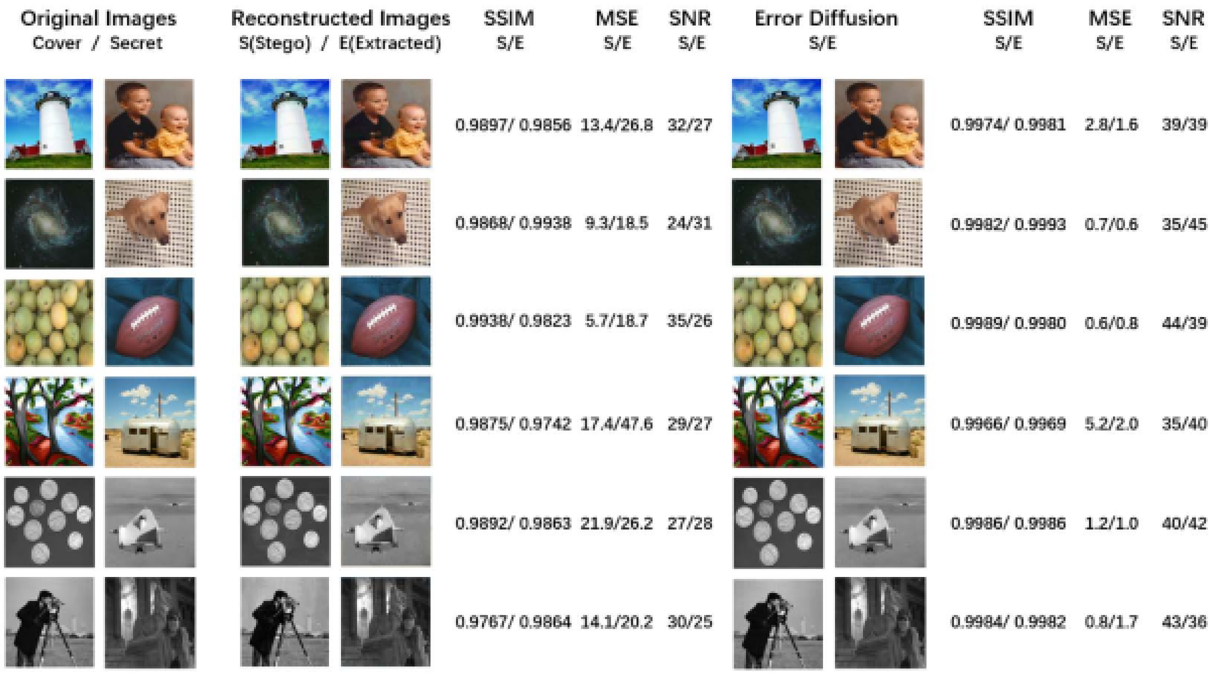


Figure 8: Six pairs of cover and secret images in the left part of this figure are selected from the Image Processing Toolbox for MATLAB, used for quantitative evaluation with SSIM [19], MSE and SNR. The middle part of this figure shows the stego and extracted secret images given by the proposed algorithm 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

[1] F. A. P. Petitcolas, R. Anderson, M. G. Huhn, "Information Hiding - A Survey," In *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999. DOI: 10.1109/5.771065. `https://ieeexplore.ieee.org/document/771065`

[2] K. U. Singh, "A Survey on Image Steganography Techniques," *International Journal of Computer Applications*, vol. 97, no. 18, pp. 10–20, 2014. DOI: 10.5120/17105-7746. `https://www.ijcaonline.org/archives/volume97/number18/17105-7746`

[3] A. Rashid, "Digital Watermarking Applications and Techniques: A Brief Review," *International Journal of Computer Applications Technology and Research*, vol. 5, no. 3, pp. 147–150, 2016. `https://ijcat.com/archives/volume5/issue3/ijcatr05031006.pdf`

[4] S. Bhattacharyya, I. Banerjee, G. Sanyal, "A Survey of Steganography and Steganalysis Technique in Image, Text, Audio and Video as Cover Carrier," *Journal of Global Research in Computer Science*, vol. 2, no. 4, pp. 1–16, 2011. `http://www.jgrcs.info/index.php/jgrcs/article/view/99`

[5] Y. Qian, J. Dong, W. Wang, T. Tan, T, "Deep learning for steganalysis via convolutional neural networks," In *Proc. of SPIE-IS&T*, vol. 9409, no. 94090J, 2015. DOI: 10.1117/12.2083479. `http://adsabs.harvard.edu/abs/2015SPIE.9409E..0JQ`

[6] C. Xia, Q. Guan, X. Zhao, Z. Xu, Y. Ma, "Improving GFR Steganalysis Features by Using Gabor Symmetry and Weighted Histograms," In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pp. 55–66, 2017. ISBN: 978-1-4503-5061-7. `https://dl.acm.org/citation.cfm?id=3083243`

[7] S. Agarwal, H. Farid, "Photo forensics from JPEG dimples," In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, 2017. DOI:10.1109/WIFS.2017.8267641. `https://www.cs.dartmouth.edu/farid/downloads/publications/wifs17.pdf`

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

Steganography and Steganalysis," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 2, pp. 142–172, 2011. https://pdfs.semanticscholar.org/3219/b0b80fe8373899f04a9f68826e94475b2c66.pdf

[9] N. Provos, P. Honeyman, "Hide and seek: an introduction to steganography," *IEEE Security & Privacy*, vol. 99, no. 3, pp. 32–44, 2003. DOI: 10.1109/MSECP.2003.1203220. https://ieeexplore.ieee.org/document/1203220

[10] J. Fridrich, M. Goljan, R. Du, "Detecting LSB steganography in color, and gray-scale images," *IEEE MultiMedia*, vol. 8, no. 4, pp. 22–28, 2001. DOI: 10.1109/93.959097. https://ieeexplore.ieee.org/document/959097

[11] Pande Gede Pradnya Jaya S.T., Bambang Hidayat, Fiky Y Suratman, "Enhanced LSB Steganography with people detection as stego key generator," In *Proceedings of International Conference on Signals and Systems (ICSigSys)*, 2017. https://ieeexplore.ieee.org/document/7967078

[12] A. M. Odat, M. A., Otair, "Image Steganography using Modified Least Significant Bit," *Indian Journal of Science & Technology*, vol. 9, no. 39, 2016. DOI: 10.17485/ijst/2016/v9i39/86878. http://www.indjst.org/index.php/indjst/article/view/86878

[13] M. M. A. Hadidi, Y. K. Ibrahim, H. K. Ali, "Data Hiding Using Least Significant Bit Approach," In *Proceedings of the 15th WSEAS International Conference on Systems*, Recent Researches in System Science, pp. 238–240, 2011. ISBN: 978-1-61804-023-7. http://www.wseas.us/e-library/conferences/2011/Corfu/SYSTEMS/SYSTEMS-37.pdf

[14] S. Baluja, "Hiding Images in Plain Sight: Deep Steganography," *Advances in Neural Information Processing Systems 30*, Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., pp. 2069–2079, 2017. https://papers.nips.cc/paper/6802-hiding-images-in-plain-sight-deep-steganograp

[15] D. L. Lau, G. R. Arce, *Modern Digital Halftoning* Second Edition; CRC Press, 2008. ISBN 9781420047530. https://www.crcpress.com/Modern-Digital-Halftoning/Lau-Arce/p/book/9781420047530

[16] R. W. Floyd, L. Steinberg, "An adaptive algorithm for spatial grey scale," In *Proceedings of the Society of Information Display*, vo. 17, pp. 75-77, 1976.

[17] M. Sakauchi, Y. Ohsawa, M. Sone, M. Onoe, "Management of the Standard Image Database for Image Processing Researches (SIDBA)," *ITEJ Technical Report*, vol. 8, no. 38, pp. 7–12, 1984. (in Japanese) https://www.jstage.jst.go.jp/article/tvtr/8/38/8_KJ00001965452/_article/-char/ja/

[18] G. Luzardo, J. Aelterman, H. Luong, W. Philips, D. Ochoa, "Real-time false-contours removal for inverse tone mapped HDR content," In *Proceedings of the 25th ACM international conference on Multimedia*, 2017. DOI: 10.1145/3123266.3123400. https://biblio.ugent.be/publication/8533776

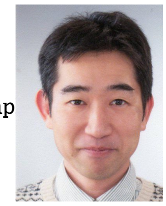
[19] Z. Wang, A. C. Bovik, R. Sheikh, E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. DOI: 10.1109/TIP.2003.819861. https://ieeexplore.ieee.org/document/1284395

[20] A. D. Ker, "A General Framework for Structural Steganalysis of LSB Replacement," In: *Information Hiding*, Barni M., Herrera-Joancomartí J., Katzenbeisser S., Pérez-González F., Eds.; IH 2005; Lecture Notes in Computer Science, vol. 3727. Springer, Berlin, Heidelberg, 2005. https://link.springer.com/chapter/10.1007/11558859_22

[21] S. Dumitrescu, X. Wu, Z. Wang, "Detection of LSB Steganography via Sample Pair Analysis," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1995–2007, 2003. DOI: 10.1109/TSP.2003.812753. https://ieeexplore.ieee.org/document/1206706

[22] A. D. Ker, R. Böhme, "Revisiting Weighted Stego-Image Steganalysis," *Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, Proc. SPIE Electronic Imaging*, , vol. 6819, San Jose, CA, pp. 0501–0517, 2008. http://www.cs.ox.ac.uk/andrew.ker/docs/ADK30B.pdf

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.