International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 5 (2017) pp. 582-590
© Research India Publications. http://www.ripublication.com

Non-fragile High quality Reversible Watermarking for Compressed PNG image format using Haar Wavelet Transforms and Constraint Difference Expansions

Junkyu Park
Department of Cyber Security, Sangmyung University,
20 Hongjimun 2-gil Jongno-gu, Seoul, Republic of Korea.

SeungIl Yu
Daolsoft Inc.,
29, Seochojungang-ro 20-gil, Seoul, Republic of Korea.

Sang-ug Kang*
Department of Computer Science, Sangmyung University,
20 Hongjimun 2-gil Jongno-gu, Seoul, Republic of Korea.
*(*Corresponding Author)*

Abstract

Most artists are reluctant to apply undesirable modifications to their original work even when the extent of change is small. In addition, they want to protect their work by marking their ownership. Many digital technologies developed for legitimate digital content distribution via the Internet utilize algorithms involving lossy features. To minimize the likelihood of incurring such features, this paper proposes a high-quality reversible watermarking method for lossless image compression of PNG images. In this method, robustness against undesired external attacks is considered for real-world usage. The data to be hidden are binary, comprising zeros and ones, and the cover image is in the compressed PNG format with a size of 512 × 512 or 256 × 256. The proposed algorithm is based on a constraint difference expansion (DE) algorithm and discrete wavelet transforms (DWTs) with Haar filters for achieving both reversibility and robustness. We successfully apply our method to various test images, and the simulation results show the superior performance of our method in terms of the high visual quality of stego-images, high data embedding capacity, and low computational complexity. Peak signal-to-noise ratio (PSNR) is used for comparing the cover and stego-images. The PSNR values of all the experimental images are approximately 60–70 dB on an average. The embedding capacity differs from image to image because the maximum capacity is embedded by applying a discriminant to ensure that overflow/underflow does not occur in an image. For the 512 × 512 size, the smallest capacity is approximately 60 bits. To measure the degree of robustness, we use the survival-ratio (SVR), which calculates the proportion of surviving bits after attacks. We generate low visible distortions, e.g., Gaussian noise, in the image to demonstrate the robustness of our algorithm. The result shows that our algorithm is robust against noise attacks within a specific range.

Keywords: Robust watermarking reversible watermarking; data hiding; PNG format; Haar wavelet transform;

INTRODUCTION

With the growth of the Internet and the development of digital image compression technologies, plagiarism of digital contents is accelerating. Consequently, interest in data hiding methods has increased owing to the high demand for copyright protection. Many image data hiding algorithms have been proposed during recent decades, with reversible watermarking attracting the most interest because of its ability to recover the cover image without information loss, a desirable characteristic for artistic work and medical applications. In this paper, the PNG image format is selected for applying a new non-fragile, high-quality reversible watermarking scheme because most artists want to preserve their original artwork without causing damage. Unlike the algorithms for other image compression standards, such as JPEG and TIFF, the PNG compression algorithm is lossless because it uses the deflate algorithm. The reversibility of the watermarking method also supports artists' desire or requirement for minimum modification to their work when using state-of-the-art digital image processing technologies. Despite the fact that an image can be perfectly recovered using a reversible watermarking algorithm, malicious attacks such as modification and recompression can still change digital contents. Therefore, a data hiding scheme must also be robust against various external attacks in order to be useful in practice. According to the data insertion position, data hiding methods for digital images are grouped into two categories: spatial domain and frequency domain. A reversible data hiding technique in the spatial domain has the characteristic that the embedded data are distributed throughout the spatial domain, usually in positions in which temporal locality is not sensitive. Many studies based on DE and histogram-shifting (HS) methods have been conducted. The first DE method, which hides data in the difference value of pixels, thereby creating an additional space by doubling or quadrupling the difference value within the scope of image content representation, was proposed by Tian [1]. Yang et al. [2] proposed an HS method that embeds data by modifying the gray-level value of a pixel referencing the zero and peak points of the image histogram.

Data embedding in the frequency domain is advantageous in that it is difficult to delete the embedded data because it is distributed throughout the image area. To embed data in the transform domain, discrete Fourier transforms (DFTs), discrete cosine transforms (DCTs), and discrete wavelet transforms (DWTs) are primarily used for converting data into the transform domain. Kim et al. [3] proposed a watermarking method using a linear feedback shift register (LFSR) in a two-level DWT domain. After determining the two-level DWT of an image, the binary watermark image is merged into a multiresolution image. The binary watermark image and the cover image are divided into blocks, and the blocks are indexed in ascending order according to the number of ones in a binary watermark image block and the variance value of a cover image block, respectively. Thus, the watermark image portion containing many ones is hidden in a complicated texture portion of the cover image. To evaluate the robustness, Kim et al. added various artificial distortions to the stego-image, such as brightness variation and low-frequency filtering. The data extraction performance depends on both the extent of attacks and the compression ratio of the lossy JPEG image compression standard. Ni et al. [4] proposed a data-embedding algorithm that divides the image into 8 × 8 blocks and uses a DE method to embed bits into the difference value between the average value per block and neighboring blocks. According to the difference value, there are four categories of data insertion; these categories are separated by thresholds. In addition, each category has several cases that determine whether an error correction coding (ECC) or data are to be hidden. Lee et al. [5] suggested a lossless visible watermarking method using the alpha channel plane of a PNG image. Pixels of the cover image replaced by the watermark are randomly inserted into the alpha channel. Kim et al. [6] proposed a robust reversible data hiding technique for the JPEG image format using average prediction DE. Its robustness results from using Bose, Chaudhuri, and Hocquenghem (BCH) codes, identical data insertion into four different images, and sophisticated selection of data insertion positions in DCT coefficients. In most of the related papers other than [6], secret watermarks are graphics such as logos or binary images because technologies for text insertion are limited by the fragile characteristics of text representation in binary code.

In this paper, we propose a non-fragile, high-quality reversible watermarking method for the compressed PNG image format using Haar wavelet transforms (HWTs) and constraint DE. Binary text data, rather than graphic data, are inserted into the cover image to sufficiently increase the payload capacity in order to accommodate copyright information. The proposed method is described in detail in Section 2. In Section 3, the simulation environment is explained, and the simulation results are shown using various numbers and figures. Section 4 provides the conclusions regarding the limitations of the proposed method and suggestions for further study.

ROBUST REVERSIBLE DATA HIDING

This algorithm comprises two principal steps: HWTs and constraint DE. When a PNG image is decompressed, a BMP image that is identical to the image before PNG compression is obtained. Although the PNG format can have up to four channels, including the alpha channel, we herein deal only with PNG format comprising three channels R, G, and B, as shown in Figure 1. Embedding data into three channels helps increase robustness because one bit of data is embedded three times into each R, G, and B channels and the three bits extracted from the decoder can be compared to determine whether they coincide. To ensure non-fragility against several simultaneous attacks, we chose to insert secret data in the frequency domain because attacks usually occur in the spatial domain. HWT is the most common and concise form of wavelet transform and has advantages in terms of computational speed and ease of implementation. In determining two-dimensional HWTs, channels are converted into four frequency forms: lo- frequency (LL), mid-frequency (LH, HL), and high-frequency (HH) sub-bands, as shown in Figure 2(b).

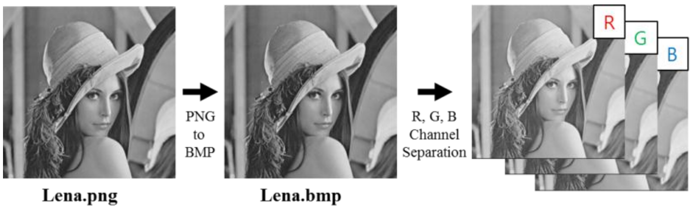


Figure 1. Separation to RGB color channels

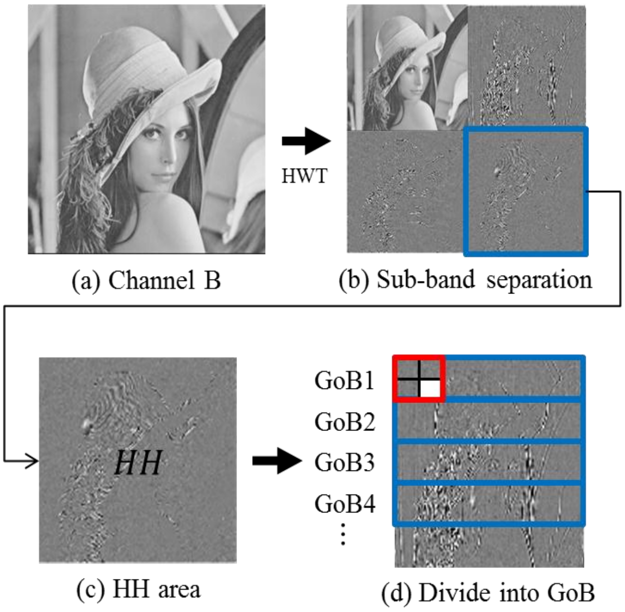


Figure 2. Regional segmentation for data hiding in the case of channel B

Determining an appropriate area to insert data is important for data hiding, especially to achieve stego-image quality. The HH region obtained by applying a high-pass filter in the horizontal and vertical directions represents the diagonal edge component of the original image. In the proposed algorithm, we choose HH because the human eye is less sensitive to high frequencies [7]. In addition, the HH region is divided into groups of blocks (GOBs); this ensures that error propagation effects caused by attackers are restricted to one GOB. A GOB comprises two 8 × 8 blocks in the vertical direction and (image width)/8

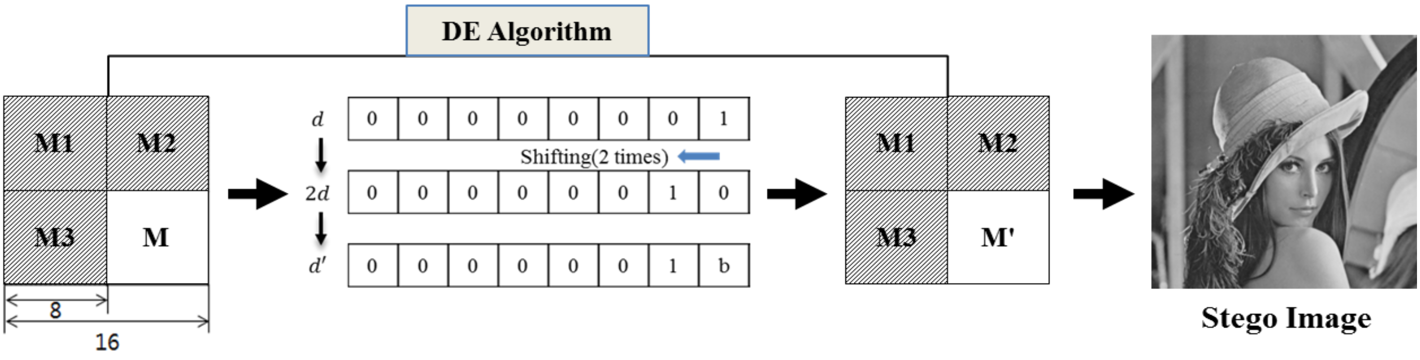


Figure 3. Constraint DE algorithm

blocks in the horizontal direction. Consequently, the size of a GOB is 16 × (image width). In addition, one set of blocks (SOB) comprises four blocks, as shown in Figure 2(d) (marked with a red line).

Data hiding algorithm

STEP 1: Read the cover image, secret data and length of data.
STEP 2: Convert the cover image to bitstream.
STEP 3: Split into R, G and B channels.
STEP 4: Transform the three channels using HWTs.

STEP 5: Embed the secret data into the transformed data. One bit is hidden into all three channels.
STEP 6: Determine the inverse HWTs.
STEP 7: Combine channels into BMP to obtain the stego-image.
STEP 8: Write the stego-image and location map.

One bit of data is hidden in the lower-right block in the SOB. Three remaining blocks are used to predict the average value of the coefficients in the target block, i.e., the white block in the SOB in Figure 2(d). An SOB is illustrated in Figure 3.

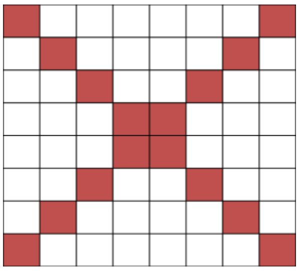


Figure 4. Selected coefficient positions in a block

Since the HH area contains the diagonal edge component of the image, the coefficients selected for calculating the average are located along the two diagonal lines of a block, as shown in Figure 4. Note that $M1$, $M2$, and $M3$ are the average values of the upper-left block, upper block, and left block of the target block, respectively. Naturally, the coefficient positions for all calculations, such as averaging and coefficient modification, are the same for all blocks. Let $\widehat{M}$ be the average value of the average values of the three remaining blocks. Secret bit b will be hidden in the difference value d between $\widehat{M}$ and M. In this case, the hiding operation can be expressed as follows:

$$d = \widehat{M} - M \tag{1}$$

$$d' = 2 \times d + b(b = 0\ or\ 1) \tag{2}$$

The expanded difference value d' is obtained by hiding the secret bit in the difference value. We can obtain a new average value using M' as follows:

$$M' = \widehat{M} + d' \tag{3}$$

$$r_i' = r_i + (M' - M) \tag{4}$$

To apply the new average value in HH, some coefficients in the target block must be changed according to Equation (4). When a bit is embedded, the same bit is redundantly embedded in all three channels with the result that the hiding capacity is reduced to one-third whereas non-fragility is increased. This process is continued along the first GOB by moving the SOB one block to the right until the end of the GOB. An SOB is not placed across the GOB boundary to ensure that the next SOB will include the first four blocks in the second GOB. If the maximum capacity of embedded data in the HH region is C_{max}, then the formula for obtaining C_{max} is

$$C_{max} = \frac{H \times (W-8)}{16 \times 8}\ (bits) \tag{5}$$

where H and W are the height and width of the HH region, respectively. To minimize the changes in the coefficient value at a particular level, a constraint DE algorithm is applied, in which secret data are inserted only in the SOB satisfying a discriminant. We set the discriminant to a zero difference between the predicted average value $\widehat{M}$ and the actual average value M. The discriminant is primarily used for avoiding overflow/underflow problems when the wavelet transform coefficients are converted to the spatial domain. Consequently, the data embedding capacity for each channel might be different.

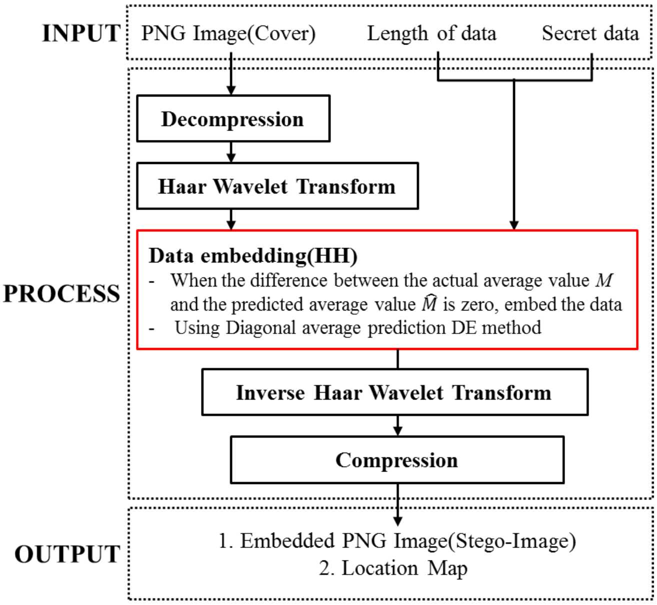

INPUT
PNG Image(Cover)
Length of data
Secret data
PROCESS
Decompression
Haar Wavelet Transform
Data embedding(HH)
- When the difference between the actual average value M and the predicted average value $\widehat{M}$ is zero, embed the data
- Using Diagonal average prediction DE method
Inverse Haar Wavelet Transform
Compression
OUTPUT
1. Embedded PNG Image(Stego-Image)
2. Location Map

Figure 5. Data embedding algorithm

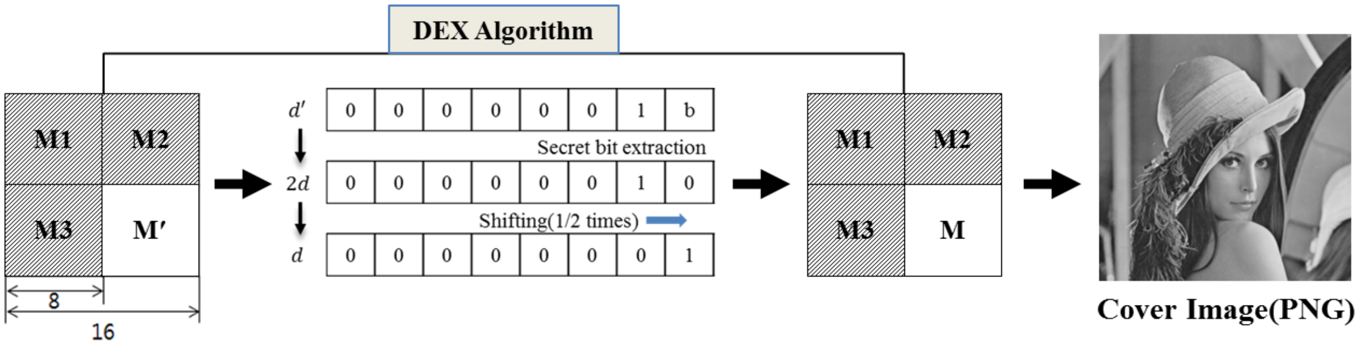


Figure 6. Constraint DEX(Data Extraction) algorithm

To determine the locations of hidden data during data extraction, a location map is created using the location information in the area satisfying the discriminant. After the insertion is completed throughout the embeddable HH coefficients, the LL, LH, HL, and HH coefficients are all converted to the spatial domain using inverse HWTs (IHWTs). The stego-image is finally acquired by integrating the embedded R, G, and B channels and compressing the BMP image to the PNG format. In summary, the proposed data embedding algorithm reads the cover image, length of the secret data, and secret data comprising zeros and ones, as illustrated in Figure 5. The result is decompressed, and then each of the three channels is transformed to the frequency domain using HWTs. Secret data are embedded in the permitted space in the HH area of each channel. When secret data are inserted, their position is recorded in the location map. Subsequently, using IHWTs, sub-bands are transformed into a channel. Finally, we combine the three channels into the BMP image and compress it to obtain the stego-image.

Data extraction and restoration algorithm

STEP 1: Read the stego-image and location map.
STEP 2: Convert the PNG stego-image to a BMP image.
STEP 3: Split into R, G, and B channels.
STEP 4: Transform the three channels using HWTs.
STEP 5: Extract secret data from the transformed coefficients.
STEP 6: Determine the inverse HWTs.
STEP 7: Combine the channels into a BMP image to obtain the cover image.
STEP 8: Write the cover image and extracted data.

The data extraction algorithm is the reverse of the data hiding algorithm. First, the stego-image that is the output of the data embedding algorithm is decompressed and split into R, G, and B channels. Then, HWTs are used for each channel to convert it to the frequency domain. The location map shows where hidden data are located in the HH area. Accordingly, embedded data can be extracted from the positions specified in the position map. The order of extraction must begin from the last SOB to preserve the causality of the average prediction.
In Figure 6, corresponding to the data extraction algorithm, M' indicates the average value of two diagonal coefficients in the 8 × 8 block. Let the predicted average value $\widehat{M}$ be the average value of M1, M2, and M3. Secret bit b is hidden in the difference value d' between $\widehat{M}$ and M' .The method of extracting embedded data is as follows:

$$d' = M' - \widehat{M} \tag{6}$$

$$b = |d'\%2|\,, b = 0 \; or \; 1 \tag{7}$$

The secret bit is extracted using the Equation (7), and the expanded difference value is divided by two in order to obtain the pre-existing difference value, as shown in equation (8):

$$d = (d' - b)/2 \tag{8}$$

The original average value of the block before embedding is obtained by subtracting pre-existing difference from the predicted average. Subsequently, the coefficients in the block are replaced with the original values, and the HH area is finally returned to its original state. Equations (9) and (10) are the formulas used to recover the original coefficients.

$$M = \widehat{M} - d \tag{9}$$

$$r_i' = r_i - (M' - M) \tag{10}$$

Using IHWTs, the HH region is converted to the R, G, and B channels along with LL, LH, and HL, and these are combined to form a BMP format image. Then, the BMP image is compressed to the PNG format. The PNG image here is expected to be the same as the PNG image before data insertion, in accordance with the reversible feature of the proposed algorithm. Since the same bits are hidden in R, G, and B, the extracted bits are also the same provided that there has been no attack on the stego-image. This enables it to be used for error detection. To summarize, the data extraction algorithm is the same as the data embedding algorithm, as illustrated in Figure 5, except for the data extraction process, which is marked with a red box in Figure 7. As inputs, the stego-image and location map are read first. The image is decompressed and converted to R, G, and B channels, which are then transformed to wavelet coefficients using HWTs.

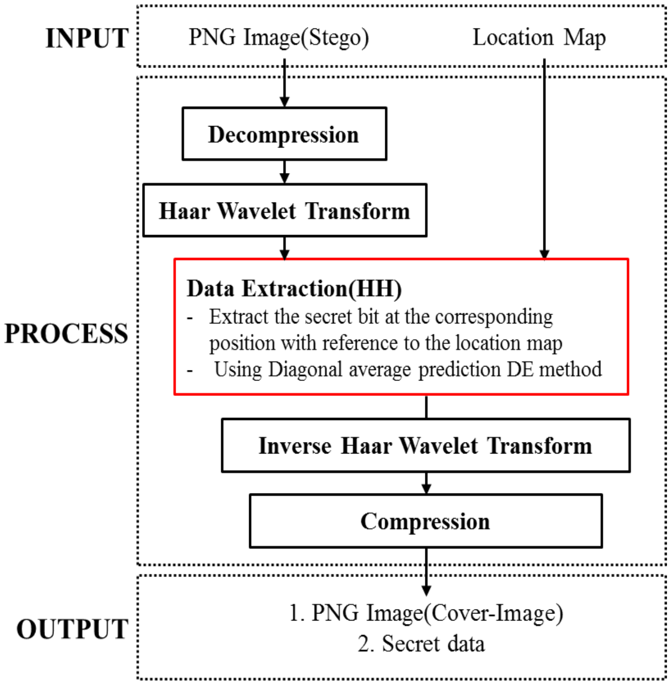

INPUT
PNG Image(Stego)
Location Map
PROCESS
Decompression
Haar Wavelet Transform
Data Extraction(HH)
- Extract the secret bit at the corresponding position with reference to the location map
- Using Diagonal average prediction DE method
Inverse Haar Wavelet Transform
Compression
OUTPUT
1. PNG Image(Cover-Image)
2. Secret data

Figure 7. Data extraction algorithm

Secret data are extracted at the corresponding position with reference to the location map using diagonal average prediction DE. When data extraction is complete, the image is expected to be the same as it was before data insertion. The existing cover image and extracted data can be obtained using IHWTs and compression.

Error detection algorithm

We embedded data into three channels and thereby helped in increasing robustness. The secret data extracted from one channel are compared with those extracted from the other channels to determine whether they coincide. If two data bits extracted from the corresponding SOBs of two channels are the same, we consider that the remaining bit from the corresponding SOB of the other channel is the same as the previous extracted two data bits even though it has a different value. When performed together with average prediction DE and frequency-domain data hiding, this algorithm improves robustness in terms of coping with a noise attack. For example, after a noise attack occurs, even if the secret bit is damaged, some part of it is recovered and the bit is preserved. The data extraction algorithm is illustrated in Figure 8.

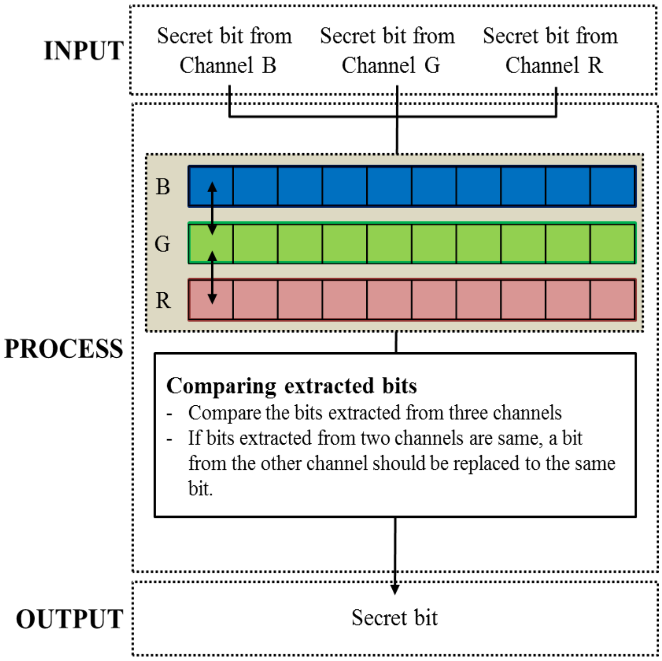
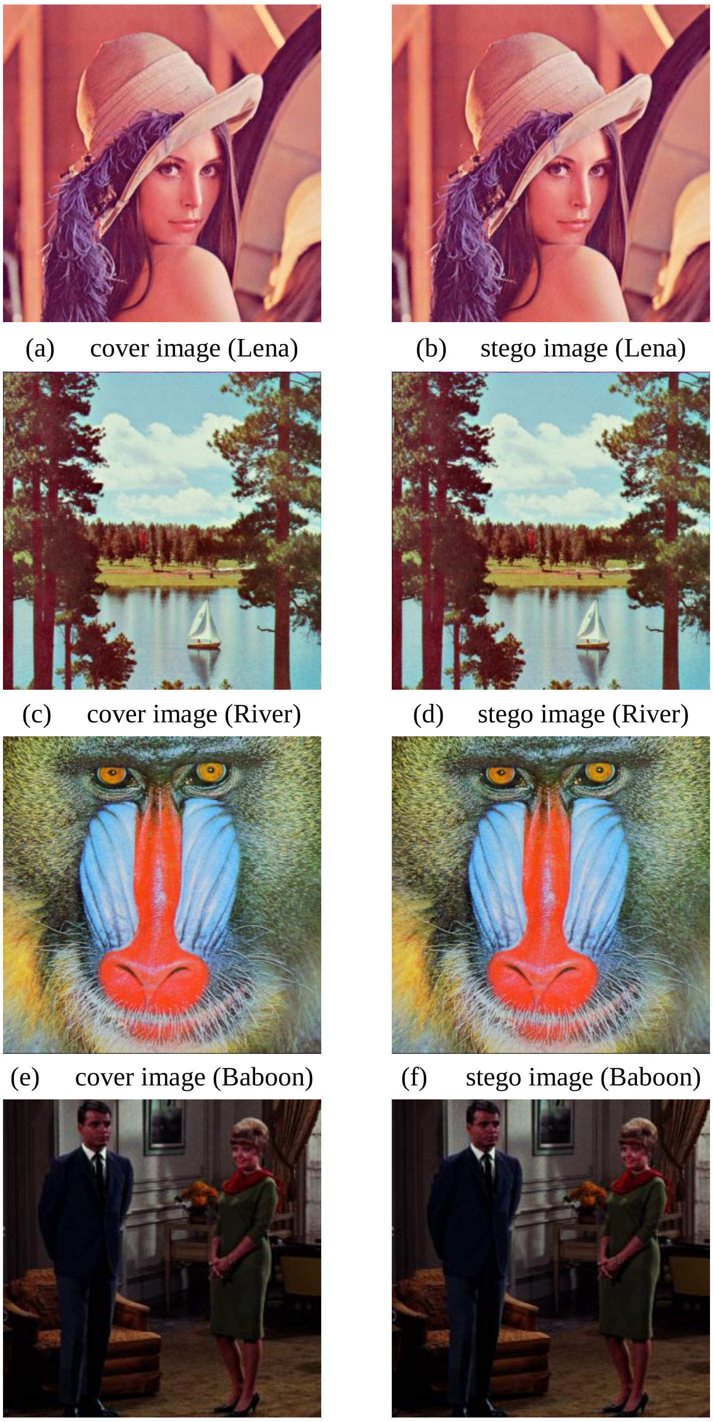

INPUT
Secret bit from Channel B
Secret bit from Channel G
Secret bit from Channel R
PROCESS
B
G
R
Comparing extracted bits
- Compare the bits extracted from three channels
- If bits extracted from two channels are same, a bit from the other channel should be replaced to the same bit.
OUTPUT
Secret bit

Figure 8. Error detection algorithm

EXPERIMENTAL RESULTS

All the experiments were conducted using 24 bpp RGB PNG compressed test images, three 512 × 512 images (Lena, River, Baboon), and three 256 × 256 images (Couple, Girl, Jelly) obtained from the USC-SIPI Image database. The secret data that we embedded were random binary data with a length of 400 and comprised zeros and ones. In addition, we used libpng, a platform-independent library based on C that contains the functions required to manipulate PNG images. This experiment aimed to analyze image quality in terms of PSNR and payload capacity and to check the degree of robustness by measuring SVR under a noise attack. The images shown on the right-hand side in Figure 9 include secret data. Distortion is not visible to the human eye even after secret data are embedded.



(a) cover image (Lena)

(b) stego image (Lena)

(c) cover image (River)

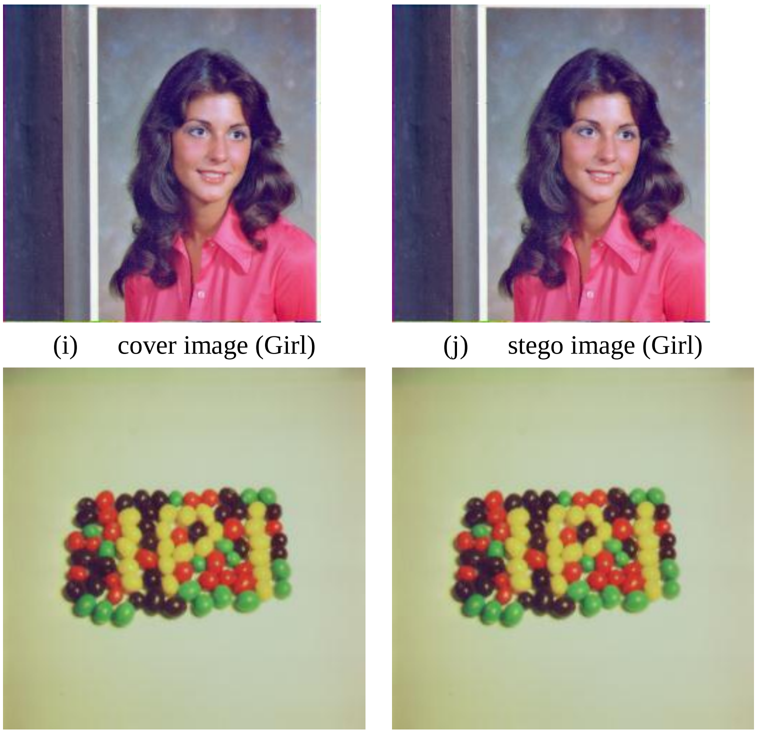
(d) stego image (River)

(e) cover image (Baboon)

(f) stego image (Baboon)

(g) cover image (Couple)

(h) stego image (Couple)



(i) cover image (Girl) (j) stego image (Girl)

(k) cover image (Jelly) (l) stego image (Jelly)

Figure 9. Comparison between the cover image and stego image

Table 1. PSNR and total payload capacity obtained from the experiment with no attack

Image	Threshold	Result
Lena (512 × 512)	PSNR(dB)	54.06
	Actual Capacity/ Total Payload Capacity	112 bits /511 bits
River (512 × 512)	PSNR(dB)	71.50
	Actual Capacity/ Total Payload Capacity	75 bits /254 bits
Baboon (512 × 512)	PSNR(dB)	72.21
	Actual Capacity/ Total Payload Capacity	61 bits /197 bits
Couple (256 × 256)	PSNR(dB)	66.72
	Actual Capacity/ Total Payload Capacity	46 bits /166 bits
Girl (256 × 256)	PSNR(dB)	66.82
	Actual Capacity/ Total Payload Capacity	48 bits /154 bits
Jelly (256 × 256)	PSNR(dB)	64.94
	Actual Capacity/ Total Payload Capacity	89 bits /280 bits

Experiment under no attack

The data to be embedded are random binary data comprising zeros and ones. Even if the embeddable space in the HH area varies depending on the discriminant set according to the proposed algorithm, we hide as many of the data as possible. This is why the payload capacity varies image by image. Another factor in determining the actual payload capacity is that the minimum capacity among the R, G, and B channels is the actual capacity because one bit of data is hidden in all three channels simultaneously. In the case of Lena, we can hide 112 bits on the B channel, 161 bits on the G channel, and 238 bits on the R channel; however, the actual capacity is 112 bits and the total payload capacity is 511 bits (= 112 + 161 + 238). **Table** 1 shows that the PSNR for this algorithm is substantially greater than that for the other data hiding algorithms. Since the PSNR for this algorithm is approximately 60dB, it can be considered to indicate that the stego-image is almost the same as the original image.

Table 2. PSNR and SVR obtained from the experiment with noise attack

Image \ Threshold		Amount of Gaussian Noise (1)	Amount of Gaussian Noise (0.5)
Lena (512×512)	PSNR (dB)	39.12	50.39
	SVR (%)	49.11	79.46
River (512×512)	PSNR (dB)	39.17	50.45
	SVR (%)	57.33	86.67
Baboon (512×512)	PSNR (dB)	39.08	50.44
	SVR (%)	44.26	67.21
Couple (256×256)	PSNR (dB)	39.29	50.44
	SVR (%)	65.22	86.96
Girl (256×256)	PSNR (dB)	39.02	50.49
	SVR (%)	62.50	83.34
Jelly (256×256)	PSNR (dB)	39.10	50.49
	SVR (%)	51.69	85.39

Experiment on noise attack

Herein, we employ a Gaussian noise attack using Adobe Photoshop CS6 on the stego-images, as shown in Figure 9, because Gaussian noise is commonly used when performing experiments on the robustness evaluation of watermarking and data hiding techniques. The selected amounts of Gaussian noise for this experiment are 0.5 and 1 per image. The SVR is 55% and 82% on an average when the amount of noise is one and 0.5, respectively. Compared with the SVR values of the 512 × 512 images, the SVR values of the 256 × 256 images is relatively high. This also indicates that the SVR is affected by the size of images as a result of HWT characteristics. In contrast to the PSNR values shown in Table 1, the PSNR values listed in Table 2 are calculated by comparing the stego-image and attacked image. The equations for obtaining PSNR and SVR are as follows:

$$PSNR = 10log\frac{255^2}{MSE}; MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I_{i,j} - I'_{i,j}\right]^2 \quad (11)$$

$$SVR(\%) = \frac{no.\ of\ survival\ bits}{Total\ embedded\ bits} \times 100 \quad (12)$$

CONCLUSTION

We proposed a non-fragile, reversible data hiding method that embeds secret data using a constraint DE algorithm in the HH domain obtained using HWTs. The secret data can be replaced with text in ASCII or UTF-8, both of which are suitable for representing copyright information. Using several attack protection methods, our scheme achieved somewhat non-fragile characteristics with regard to the Gaussian noise attack. The average PSNR for the stego-images in total was approximately 60–70 dB, which indicates very high visual quality. Constraint DE contributes substantially to this high fidelity; however, the extraction scheme using a location map requires further study. Its size can be reduced using compression or clever position identification. In addition, we expect to be able to develop a new wavelet transform to accurately predict positions.

ACKNOWLEDGMENT

This research project was supported by Ministry of Culture, Sports and Tourism (MCST) and by Korea Copyright Commission in 2016 (2014UCI9500).

REFERENCES

[1] Tian, J., 2003, “Reversible Data Embedding Using a Difference Expansion,” IEEE T. Circ. Syst. Vid. 13(8), pp. 890-896.

[2] Ni, Z., Shi, Y-Q., Ansari, N., and Su, W., 2006, “Reversible Data Hiding,” IEEE T. Circ. Syst. Vid. 16(3), pp. 354-362

[3] Kim, T. J., Hong, C. S., and Hwang, J. H., 2008, “A Wavelet Based Robust Logo Watermarking Algorithm for Digital Content Protection,” J. Internet Comput. Serv., 9(1), pp. 33-41.

[4] Ni, Z., Shi, Y-Q., Ansari, N., Su, W., Sun, Q., and Lin, X., 2008, “Robust Loseless Image Data Hiding Designed for Semi-Fragile Image Authentication,” IEEE T. Circ. Syst. Vid. 18(4), pp. 497-509.

[5] Lee, C-W., and Tsai, W-H., “A New Loseless Visible Watermarking Method via the Use of the PNG Image,” 2012

[6] Kim, H. J., Yu, S. I., and Kang, S. U., 2017, “Robust Reversible Data Hiding Technique toward Modification and Recompression in JPEG Images using Average Prediction Difference Expansion,” Int. J. Appl. Eng. Res., 12(1), pp. 110-118

[7] Zagade, S., and Bhosale, S., 2014, “Secret Data Hiding in Images by using DWT Technique’s”, Int. J. Eng. Adv. Technol., 3(5), pp. 230-235