

## Robust Reversible Data Hiding Technique toward Modification and Recompression in JPEG Images using Average Prediction Difference Expansion

**Hyunjung Kim**

*Department of Computer Science, Sangmyung University,  
20 Hongjimun 2-gil Jongno-gu, Seoul, Republic of Korea.*

**SeungIl Yu**

*,Daol soft 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

Since digital images are one of the most popular types of content, their distribution must include copyright protection provisions. Protection after creation is normally carried out by marking the copyright on the image or identifying the image using watermarking or fingerprinting techniques. However, the addition of a watermark damages the original image and it is thus not suitable for information that must be frequently updated, such as the distribution information. Digital fingerprinting also cannot be used to insert any type of distribution information. In contrast, the technique proposed in this study can reversibly hide about 42 bytes of distribution information in the image, and it can also be constantly updated without damaging the original image. Furthermore, it is robust against recompression, noise, and modification attacks that frequently occur in the distribution processes. The proposed technique secures the data distribution by adding robustness and masking through BCH (31,21,2) error correcting coding technology, and statistical values, and adds reversibility by using the difference expansion method. An analysis of the experimental results shows that the proposed technique exhibits 100% data restoration power in recompression attacks using the same quality factor (QF), maintains its robustness within a certain range against recompression attacks using a different QF, and is robust within a certain range against noise and image modification attacks.

**Keywords:** Robust data hiding; Reversible data hiding; JPEG; Recompression; BCH code.

### INTRODUCTION

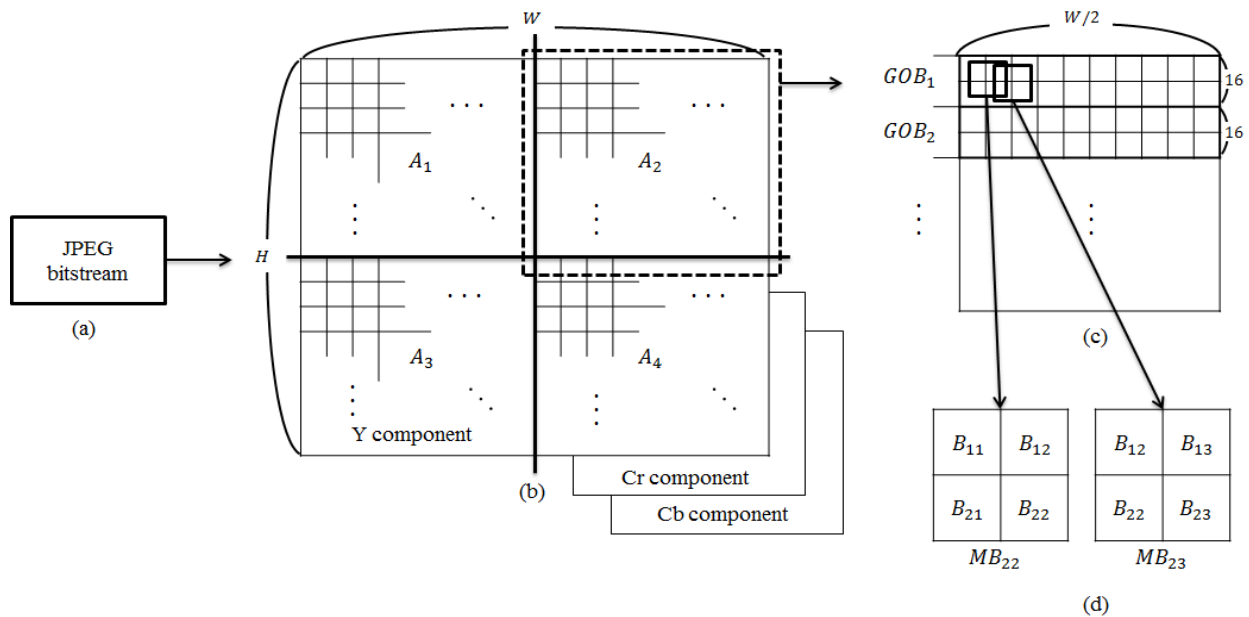
Digital content is moved and copied at high speed on the internet, and thus it is not easy to determine how often it is used or distributed. At the same time, content providers want to know the distribution status of their content for financial or other reasons. For example, royalty calculations are dependent on the popularity of the content, and distribution related information is required for financial reasons. On the other hand, providers of free content generally calculate the

popularity of their content based on the corresponding number of hits or shares. To this end, content distribution information, such as the copyright holder, producer, and content identifiers are hidden inside the content and used to track the distribution, validate copyright claims, and determine whether content is being used illegally. In most cases, this distribution information requires at least 50 bytes, and must be impervious to damage, extraction, or unauthorized modification because the hidden information must be present and correct in order to accurately identify content that exists on the internet and to determine its distribution status.

Watermarking and fingerprinting techniques are commonly used to automatically identify images. Watermarking allows users to visibly or invisibly insert relevant information into the image and then to identify images using this information when necessary [1][2]. In general, fingerprinting extracts the innate features of images, saves them in the form of a database, and mutually compares them to the features of the image that must be identified [3][4]. These two techniques have been implemented in the form of algorithms that are relatively robust against non-geometrical transformations, such as Gaussian noise, brightness changes, and geometrical transformations, such as enlargement, reduction, rotation, and aspect ratio changes. However, in terms of the quantity of information that can be inserted without damage to the original, these techniques cannot be used to embed the necessary distribution information without damaging the original image, which is the main objective of this study. Considering the amount of information that can be embedded, fingerprinting is inappropriate since it can only identify the image but cannot be used to insert information, and watermarking is inappropriate because it inserts information in the native image format instead of a coded text format, and is therefore not capable of being used to embed the distribution information. In terms of damage, fingerprinting does not cause any damage to the original, but watermarking inevitably causes damage, and the severity of the damage becomes more severe if the algorithm is made more robust against attacks. The distribution information must be updated constantly based on copyright exchanges and image shifts. If

damage is caused to the image by frequent modifications of the watermarking, the image quality is degraded and the

meaning it was intended to convey may be distorted.



**Figure 1:** Regional segmentation method for data dispersion and hiding in the JPEG bitstream

Therefore, considering the reality that images are distributed in compressed form, this study proposes an algorithm that can be used to hide a sufficient amount of information in the JPEG format that is robust against JPEG recompression and partial content modification that may naturally occur in image content distribution, and is without damage to the original compressed image. This paper is structured as follows. In Section 2, current robust reversible data hiding techniques are analyzed. In Section 3, the robust reversible data hiding technique proposed in this study is explained. In Section 4, the result of an experiment to assess the robustness of the proposed technique against image modification and recompression, and the distortion of the stego image due to data hiding. Conclusions are provided in Section 5.

### ANALYSIS OF ROBUST REVERSIBLE DATA HIDING TECHNIQUES

It is easy for anyone to rotate, compress, or change the brightness of image content, and thus there is an increasing need for robust reversible data hiding methods. Reversible data hiding techniques hide data in the cover image without causing damage to the image, consist mostly of methods that hide data in the pixels. Vleeschouwer et al. [5] is the first to propose a semi-robust data hiding technique using patchwork theory and modulo-256 addition. However, the boundary effect caused by the modulo operation in this method creates salt-and-pepper noise in stego images. To resolve this issue and increase the robustness, Ni et al. [6] used a BCH code to first hide the data in the average of the differences of adjacent pixels with little change in the JPEG compression, and then to restore the bit errors that intentionally occurred when the data was being hidden to ensure reversibility. Zeng et al. [7] ensured robustness by maintaining a certain distance between the bit-0 zone in which the hidden data consists of is 0 bits,

and the bit-1 zone in which the hidden data consisted of 1 bits, so that the data was robust against JPEG compression effects, and prevented the two zones from overlapping with each other even if there were some changes to the pixels after JPEG compression. Yang et al. [8] proposed a method to secure robustness in the frequency domain using a coefficient shifting algorithm in the Integer Wavelet Transform (IWT) domain. Thabit et al. [9] suggested a method of embedding the data bits by modifying the differences among averages of the Slantlet transform (SLT) coefficients that were in the high frequency domain by converting the non-overlapping block of the host image using an SLT matrix. In [10], a discrete wavelet transform (DWT) algorithm was used for the transformation, the singular value was decomposed using the singular value decomposition (SVD) algorithm for each sub-band (LL, LH, HL, HH), and then the data was hidden in that value. However, the methods suggested by [6-10] are based on the premise that the data is hidden and the robustness is secured by changing the pixels themselves, and the hidden data is a binary image. In terms of a binary image, the image can often still be recognized even if some of the data is lost, which means that this method basically robust against attacks; however, not much data can be hidden. Moreover, when the data is hidden in pixels, it is difficult to reflect the reality of distributing images in compressed form.

Many studies have been conducted on hiding data reversibly in the quantized DCT (QDCT), and this technique is applicable to compressed JPEG bitstreams even though robustness has not been considered. Chang et al. [11] suggested a method of hiding data if there were two or more continuous zeros in the radio frequency component of the  $8 \times 8$  QDCT block. Xuan et al. [12] determined the optimum threshold to minimize the increase in the file size of the image due to inefficiencies in the entropy coding that increase after hiding, and data is only hidden in the QDCT coefficients in

the low and intermediate frequency domains where there will be the least distortion of the image. If the frequency of the QDCT is in the range of  $[-T, T]$ , the QDCT coefficients are  $h_0 = [a_1, a_2, a_3, \dots, a_{2T+1}]$ , the pairs of neighboring frequencies are histogram pairs expressed as  $h = [a_i, a_{i+1}]$ , and data can be hidden here. If  $a_{i+1}$  is 0 and  $a_i$  is a positive integer that is not 0, approximately  $m$  bits of data can be hidden here. Wang et al. [13] used a method that reduced the table value by dividing the quantization table value in the mid-range frequency coefficient of the QDCT block by a constant  $k$ , and multiplied the QDCT coefficient by  $k$  to identify space up to a  $k$ -array ( $0 \sim k-1$ ) in size in which to hide

data. Here, the value of the DCT coefficient is the product of multiplying the quantization table value by the QDCT coefficient, and thus the quality of the stego image with reversible data hiding was improved under the condition of using the quantization table value that was divided by  $k$ . Huang et al [14] suggested the method of setting the peak bin as the place in which to hide data as -1 and 1 among QDCT coefficients using the histogram shifting algorithm, which reduced the increase in the file size of the hidden image since the data was not hidden in the zeroes.

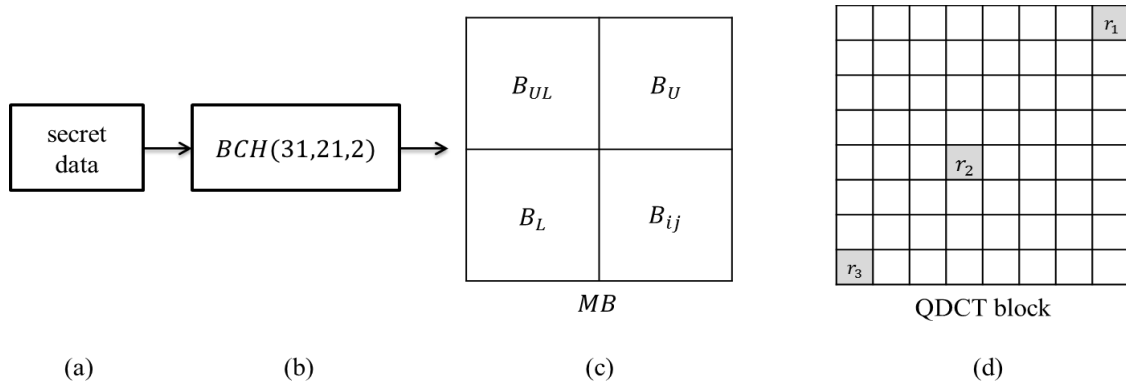


Figure 2: Detailed structure of macro block hiding data in macro block and QDCT block

This study suggests reversible data hiding algorithms that are robust against changes in the QDCT value and allow image modification and recompression while hiding data in the QDCT.

### A PROPOSED ROBUST REVERSIBLE DATA HIDING TECHNIQUE

In the proposed technique, the distributor is able to hide distribution information in text format in the JPEG bitstream with lossless restoration of the original JPEG bitstream. The watermarking technique hides the data in the binary image format and thus causes slight errors in the event of attacks. When the hidden image is damaged and then restored, the damage can be perceived by human eyes. For example, if the hidden data is “We go to school” and the data restored due to error is “We ggo to school,” it is clear to a reader that that “ggo” must be modified to “go” in terms of context. This feature is used to study diverse methods that are robust against errors. Of course, this method is not as robust as when hiding data in the image format. Moreover, if only a JPEG bitstream is realistically available for users or distributors, and it is used to deliver attacks, such as reduction, rotation and noise, the recompression process is absolutely necessary. Also, JPEG is a lossy compression method and thus the QDCT value and bitstream will change even if there is only recompression without any attack at all. Therefore, this study suggests robust algorithms to prevent the most common JPEG recompression and modification attacks that actually occur.

### Data hiding algorithm

Since the human eye is less sensitive to changes in brightness than it is to changes in color, the data is hidden in the QDCT coefficients of the Y component, which is the brightness component in the JPEG bitstream. The entire process is described in Figure 1, and is as follows.

Entropy coding is performed on the JPEG bitstream to obtain the Y component QDCT value in the size of  $W \times H$ , as shown in Figure 1 (b), and it is then divided into four domains, namely  $A_1, A_2, A_3,$  and  $A_4$ , with the size of  $W / 2 \times H / 2$ . In each domain, the same data is hidden four times for the purpose of redundancy so that if hidden data cannot be restored due to image modifications or recompression errors in one domain, it is more likely that this data can be restored using the data stored in the same place of another domain. Each domain is divided into several Group Of Blocks (GOBs) with size  $W / 2 \times 16$ , as shown in Figure 1 (c), and one GOB is divided into overlapping macro blocks with the size of  $16 \times 16$ , as shown in Figure 1 (d), each of which is referred to as  $MB_{ij}$ . Here,  $i, j$  are the locations of the block used to hide data. Macro blocks can only be formed within a single GOB, and they consist of a total of four blocks including  $B_{ij}$  to hide data. There are three related blocks:  $B_{UL}$  in the upper left,  $B_L$  to the left, and  $B_U$  above. The macro blocks are overlapped beginning with the first macro block in one GOB and moving to the right one block at a time. The suggested technique hides one bit of data in one MB. If the data is hidden using the ASCII code, one 1-byte of text can be hidden in 8 MBs, and  $W / (2 \times 8) - 1$  bits of data in one GOB. The capacity of the entire image is given by

Equation (1).

$$C = \left(\frac{W}{16} - 1\right) \times \frac{H}{32} \text{ (bits)} \quad (1)$$

The method of hiding data in the MB is explained in Figure 2. As shown in the figure, the average  $M$  of the QDCT coefficients  $r_1, r_2, r_3$  in a specific location is obtained for  $B_{UL}, B_L, B_U$ , based on Equation (2), which are referred to as  $M_{UL}, M_L, M_U$  respectively. The average is used because, even if the coefficients change due to attacks, the hidden data may still be extracted without errors if the range of the change in the  $s$  values is  $-2 \leq s \leq 2$ . Of course, if there are many coefficients used to obtain the average, it is more likely that the errors will be corrected, but the cost of the quality deterioration of the stego image must also be considered. This is because, as shown in Equation (5), the distortion due to data hiding affects more QDCT values. Moreover, because  $r_1, r_2,$  and  $r_3$  are in the intermediate frequency range, the distance between them is set up to be as far from one another as possible for the following reasons. First, the image data is condensed in the low frequency band and is thus sensitive to even slight changes. In the high frequency range, changes have a great impact on the compression rate. Second, adjacent coefficients change when changes are observed in the QDCT values during recompression.

Using Equation (3), the average  $\widehat{M}_{ij}$  is obtained, which is referred to as the predictor of the actual average  $M_{ij}$  of the block used to hide this data.

$$M = \lfloor \sum_{i=1}^3 r_i / 3 \rfloor \quad (2)$$

$$\widehat{M}_{ij} = \lfloor \frac{M_{UL} + M_L + M_U}{3} \rfloor \quad (3)$$

Equation (4) is used to obtain the difference  $d_{ij}$  of the actual value  $M_{ij}$  and predictive value  $\widehat{M}_{ij}$ . One bit of empty space is created by shifting  $d_{ij}$  one bit to the left, and Equation (5) is used to hide the one bit of data  $b$ , and the new difference  $d'_{ij}$  is created.

$$d_{ij} = \widehat{M}_{ij} - M_{ij} \quad (4)$$

$$d'_{ij} = 2d_{ij} + b, \quad b = 0 \text{ or } 1 \quad (5)$$

Equations (6) and (7) are used to obtain the new average  $M'_{ij}$  of the block to hide the data using  $d'_{ij}$  and the new QDCT values  $r'_1, r'_2,$  and  $r'_3$ , respectively.

$$M'_{ij} = \widehat{M}_{ij} + d'_{ij} \quad (6)$$

$$r'_k = r_k + (M'_{ij} - M_{ij}), \quad k = 1, 2, 3 \quad (7)$$

This process is applied to all of the images, and the new QDCT value is used for the entropy coding to create the stego bitstream. Before hiding secret data, the data to which BCH coding has been applied, as shown in Figure 2 (b), is hidden, which enables detection and restoration of the location where the error occurred, thereby increasing robustness of the algorithm. If BCH( $n, k, t$ ) is applied, the capacity is reduced to  $C_{BCH} = \lfloor C/n \rfloor \times k$  bits. For example, a  $512 \times 512$  image can be used to hide a maximum of 496 bits or 62 characters, or 42

characters when the BCH (31,21,2) is applied. The sample image shown in Figure 4 has these characteristics. Figure 3 is a diagram showing the method of creating the stego JPEG bitstream by hiding data in the overall JPEG bitstream, and the secret data extraction and JPEG bitstream recovery and restoration process.

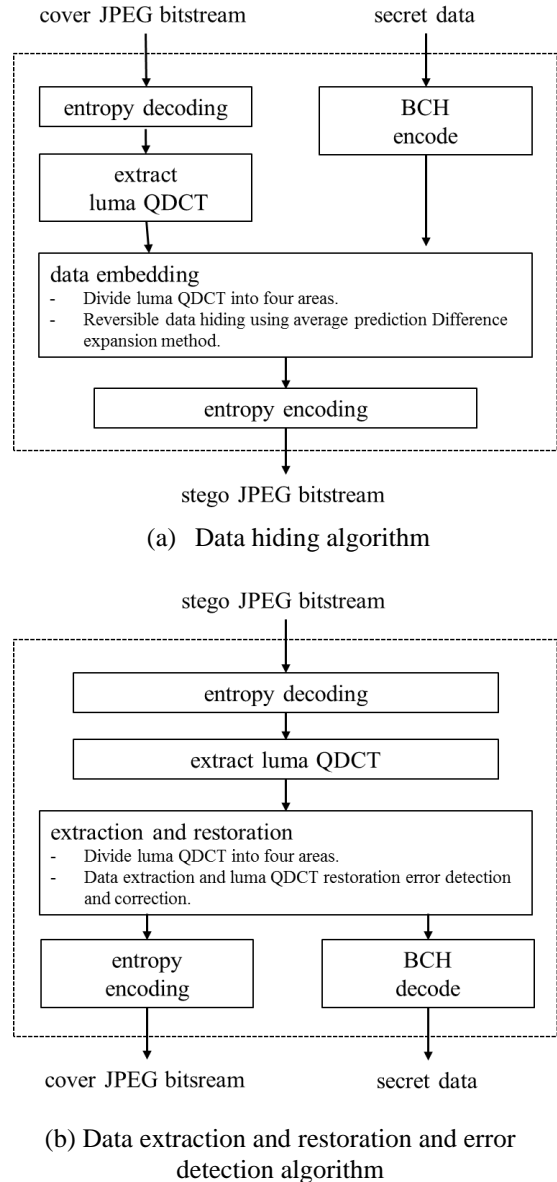


Figure 3: Diagram of the proposed scheme

#### Data extraction and restoration algorithm

The algorithm used to extract hidden data and reversibly restore the image is the inverse of the algorithm used to hide the data in the first place. After entropy decoding of the stego JPEG bitstream, the predicted average  $\widehat{M}_{ij}$  is obtained using the average  $M'_{ij}$  of the QDCT coefficients of the block to extract data and the surrounding blocks  $M_{UL}, M_L, M_U$ , and  $d'_{ij}$ , which is the difference between  $M'_{ij}$  and  $\widehat{M}_{ij}$ , and is calculated using Equation (8). Equation (9) is used to perform modulo-2 operation of  $d'_{ij}$  and the absolute value is added to

extract the hidden data. If  $d'_{ij}$  is divided by 2, as shown in Equation (9), the original difference  $d_{ij}$  can be obtained.

$$d'_{ij} = \widehat{M}_{ij} - M'_{ij} \quad (8)$$

$$b = |d'_{ij} \bmod 2| \quad (9)$$

$$d_{ij} = \lfloor \frac{d'_{ij}}{2} \rfloor \quad (10)$$

After obtaining the original  $M_{ij}$  by adding  $d_{ij}$  and the predicted average  $\widehat{M}_{ij}$ , the original  $r_k$  is restored by subtracting the difference of  $M'_{ij}$  and  $M_{ij}$  from  $r'_k$ .

$$M_{ij} = \widehat{M}_{ij} + d_{ij} \quad (11)$$

$$r_k = r'_k - (M'_{ij} - M_{ij}), k = 1,2,3 \quad (12)$$

**Error detection algorithm**

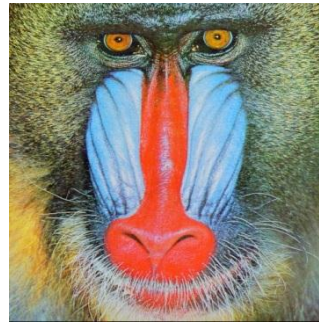
The following is the process to detect and correct the errors after extracting the data from the entire image. Suppose that the data extracted from the four domains  $A_1, A_2, A_3, A_4$  is  $BA_1, BA_2, BA_3, BA_4$ . If one bit extracted from the macro block in the same position is  $b_1, b_2, b_3, b_4$ , the number of zeroes  $num\_zero(b_1, b_2, b_3, b_4)$  and the number of ones  $num\_one(b_1, b_2, b_3, b_4)$  are obtained, and the final bit  $final\_b$  is selected as zero if  $num\_zero$  is equal to or greater than  $num\_one$  or one if  $num\_one$  is greater, as shown in Equation (13).

$$final\_b = \begin{cases} 0, & \text{if } num\_zero \geq num\_one \\ 1, & \text{if } num\_zero < num\_one \end{cases} \quad (13)$$

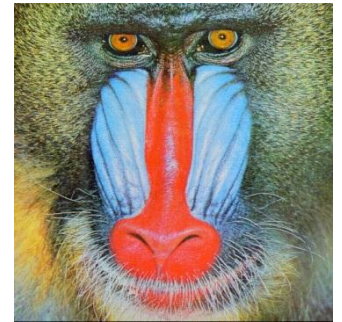
After obtaining the  $final\_b$  in each position, the BCH encoded bitstream is created, and a maximum of two errors are corrected through BCH decoding and ultimately the secret data is restored.

**EXPERIMENTAL RESULTS**

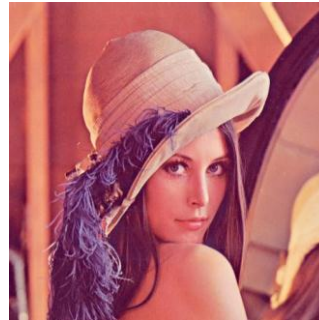
For the experiment, images were compressed and restored by mostly using  $QF = 75$ , where  $QF$  is the default Quality Factor that is applied in the commonly used JPEG player. A randomly created bitstream is used as the secret data. For the cover image, 512 X 512 size color images titled Airplane, Baboon, Lena, Peppers were used as shown in Figures. 4(a), 4(c), 4(e), and 4(g). Stego images in which 42 characters ( $C_{BCH} = 336 \text{ bits}$ ) of secret data were inserted using the proposed method are shown in



(c) cover image (baboon)



(d) stego image (baboon)



(e) cover image (Lena)



(f) stego image (Lena)



(g) cover image (Peppers)



(h) stego image (Peppers)

**Figure 4:** Cover image and stego image(total embedding capacity 62 bytes, secret data 42 bytes with BCH(31,21,2))

Figure 4(b), 4(d), 4(f), and 4(h). To the naked eye, the original image and data are hidden and it is difficult to find out the difference with the distorted image.

**Experiment about recompression attacks using the same QF**

The JPEG bitstream requires the process of recompression if a certain attack is made using the image editor, as shown in Figure 5. If the image is recompressed using the same  $QF$  without making any kind of modification, another bitstream is created. If the process of restoring the compressed bitstream and recompressing it, which is restored again and recompressed again, is repeated a maximum of 100 times, the same bitstream is created, but before then, it is likely that the JPEG bitstream may be created differently even if there is no attack [15]. Table 1 is the result of measuring the image quality according to the JPEG recompression attack on stego images and the restoration robustness of the hidden data. The



(a) cover image (Airplane)



(b) stego image (Airplane)

peak to noise ratio (PSNR) is used to determine the image quality, and the survival rate (SVR) of Equation (14) is used as a measure to assess the robustness of the hidden data.

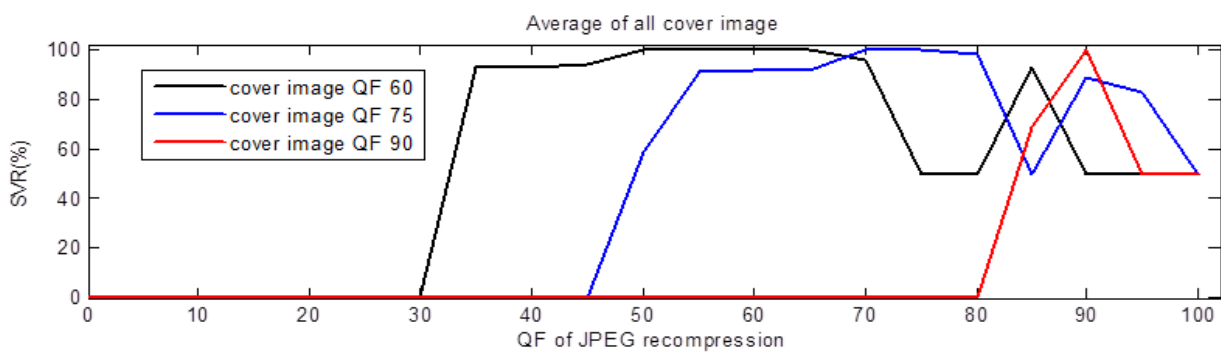
$$SVR(\%) = \frac{\text{no. of survival bits}}{c} \times 100 \quad (14)$$

The PSNR of the stego image that was not attacked was at least 33.89 dB and showed a high image quality. The SVR was 100%. When recompression was performed multiple

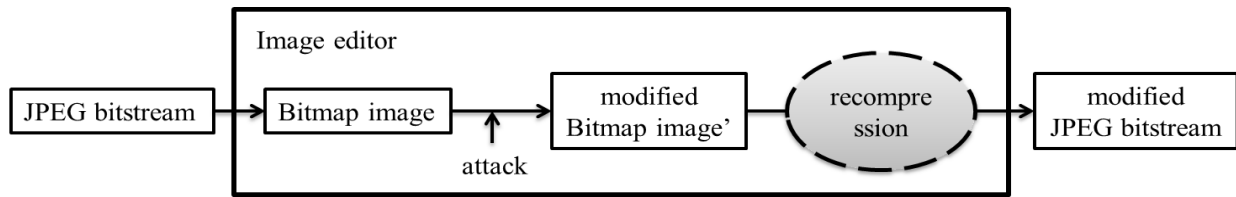
times with the same QF = 75, the bitstream was maintained consistently after three times for the Airplane, seven times for the Baboon, eight times for Lena, and eight times for the Peppers. Therefore, 11 recompressions were done repeatedly in the experiment, which is shown in Table 1. Based on the results, the PSNR decreased by 0.11 dB on average, and the SVR was 100% in recompression attacks, thereby proving that the proposed algorithms are robust.

**Table 1:** Robustness measured when there were multiple recompression attacks in the same QF value (QF = 75)

Image	No. of recompression attacks	PSNR (dB)	SVR (%)
Airplane	no attack	36.10	100
	1	36.10	100
	3	36.10	100
	5	36.10	100
	7	36.10	100
	11	36.10	100
Baboon	no attack	33.89	100
	1	33.88	100
	3	33.87	100
	5	33.87	100
	7	33.87	100
	11	33.87	100
Lena	no attack	36.41	100
	1	36.40	100
	3	36.39	100
	5	36.39	100
	7	36.39	100
	11	36.39	100
Peppers	no attack	36.43	100
	1	36.29	100
	3	36.13	100
	5	36.04	100
	7	36.03	100
	11	36.02	100



**Figure 5:** SVR based on the QF change in recompression when the cover image QF was 60, 75, and 90



**Figure 6:** JPEG bitstream attack flow chart

**Experiment about recompression attacks using different QFs**

If cover images are created using QF values of 60, 75, and 90 and they are recompressed using different QF values, this results in the creation of different bitstreams. Figure 6 shows the results of the experiment regarding the robustness against these attacks in a graph of the average SVR for the different recompression QF values. If the cover image was created with QF = 60, the secret data can be restored by recompression at a QF = 35–70. If QF = 75, it can be restored by recompression at a QF = 55–80, and at QF = 90, it is only robust when the recompression QF = 90.

**Experiment about noise attacks**

This experiment tests the robustness when noise attacks are made to the test stego image created with a QF = 75. Salt & pepper noise and speckle noise were used for the noise attacks, and the robustness and image quality were measured based on the level of noise by varying noise density  $v$ . The noise level increases as the  $v$  value increases, and the value is between zero and one. The experimental results are shown in Table 2, which shows that both the salt & pepper noise and speckle noise are 100% robust if the noise level is low with  $v = 0.001$ . If  $v = 0.005$ , the SVR is at least 88%, except for the case in which the speckle noise attack was made to the Airplane image, and thus the decoding of the restored distribution information was mostly possible. Moreover, if the PSNR was 30 or below, the image quality was considered to be much deteriorated, and the SVR reached 69.05% only when the PSNR was 28.90 dB.

**Experiment about modification attacks**





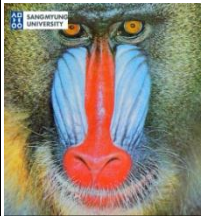
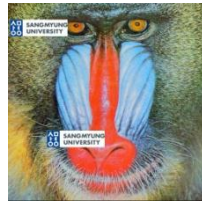
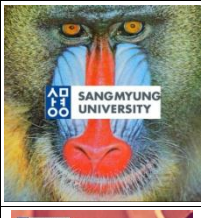
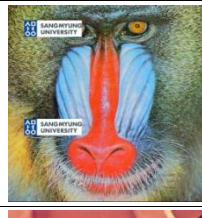



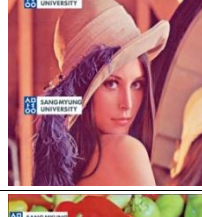




In the case of image modification attacks, it was assumed that the original images were modified and used when they were distributed. This experiment was conducted to determine how robustly the distribution information could be restored. The experiment was conducted for the four cases based on the modified location. First, the SVR was measured by inserting a 150 x 43 mark in only  $A_1$  of the stego image. Second, the same mark was inserted in  $A_1$  and  $A_3$ . Here, the location where the mark was inserted only partially overlaps in  $A_1$  and  $A_3$ . For example, the upper-left corner of the mark is located at (4, 43) in  $A_1$  and at (100, 330) in  $A_3$ . Third, a 300 x 86 mark was

inserted at the center of the stego image so that there was no overlapping in all areas of  $A_1, A_2, A_3, A_4$ . Lastly, a 150 x 43 mark was inserted in  $A_1$  and  $A_3$  in a location where it is completely overlapping: (41,40), (41, 296). Table 3 shows the results of the experiment, the modified images, and the SVR measurements.

**Table 2:** Robustness performance analysis for the noise attack

image	Noise Attack		PSNR (dB)	SVR (%)
	type	v		
Airplane	Salt & Pepper	0.001	34.28	100
		0.005	30.01	88.10
	Speckle	0.001	34.47	100
		0.005	28.90	69.05
Baboon	Salt & Pepper	0.001	32.97	100
		0.005	30.11	98.21
	Speckle	0.001	33.36	100
		0.005	30.62	100
Lena	Salt & Pepper	0.001	34.51	100
		0.005	30.44	94.94
	Speckle	0.001	35.60	100
		0.005	31.98	97.62
Peppers	Salt & Pepper	0.001	34.36	100
		0.005	30.06	83.63
	Speckle	0.001	35.35	100
		0.005	31.30	89.29

**Table 3:** Robustness performance analysis according to image modification attack

image	modification position	attack image	SVR (%)	modification position	attack image	SVR (%)
Airplane	$A_1$		100	$A_1, A_3$ (partially overlapping position)		99.11
	$A_1, A_2, A_3, A_4$ (difference position)		100	$A_1, A_3$ (same position)		98.21
Baboon	$A_1$		100	$A_1, A_3$ (partially overlapping position)		100
	$A_1, A_2, A_3, A_4$ (difference position)		100	$A_1, A_3$ (same position)		97.32
Lena	$A_1$		100	$A_1, A_3$ (partially overlapping position)		99.11
	$A_1, A_2, A_3, A_4$ (difference position)		97.92	$A_1, A_3$ (same position)		97.62
Peppers	$A_1$		100	$A_1, A_3$ (partially overlapping position)		100
	$A_1, A_2, A_3, A_4$ (difference position)		100	$A_1, A_3$ (same position)		98.21



In the first modification attack, the attack was performed on only the data hidden in one partition, and thus the SVR shows 100% performance in all images. The SVR was also at least 99% in the second modification attack, and thus the data was recognizable enough when restored to the text format. In the third modification attack, all hidden data could be restored 100% except for Lena. The fourth modification attack showed the lowest SVR due to the attack on the same location in two domains, but the error bit was less than 3% and thus it is robust against modification attacks.

## CONCLUSION

This study proposed a robust reversible data hiding technique against the modification and recompression of JPEG images using an expansion of average prediction difference. Unlike watermarking or fingerprinting techniques, this technique is able to hide the distribution information without damaging the original image. Thus, it is robust against attacks that are highly likely to occur when distributing a sufficient amount of data, and is therefore the most suitable technique to address the realities of the distribution of images. However, like other techniques, as the amount of hidden data increases, the robustness decreases. In addition, unlike the watermarking technique that hides images, this technique hides text, which causes the robustness to deteriorate. In particular, it is vulnerable to geometric attacks, such as enlargement and reduction. However, it is quite robust against recompression and image modification attacks that frequently occur in image distribution settings, and thus can be applied to various fields, such as the tracking of image content and usage monitoring. In the future, it is necessary to conduct research on algorithms that are robust against more diverse attacks while hiding a greater amount of data.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, 2010, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, 90(3), pp. 727-752,
- [2] Frédéric Lusson, Karen Bailey, Mark Leeney, Kevin Curran, 2013, "A novel approach to digital watermarking, exploiting colour spaces," *Signal Processing*, 93(5), pp. 1268-1294,
- [3] Xiushan Nie, Ju Liu, Qian Wang, Wenjun Zeng, 2015, "Graph-based video fingerprinting using double optimal projection," *Journal of Visual Communication and Image Representation*, 32, pp. 120-129,
- [4] Xinwei Li, Baolong Guo, Fanjie Meng, Leida Li, 2011, "A novel fingerprinting algorithm with blind detection in DCT domain for images," *AEU - International Journal of Electronics and Communications*, 65(11), pp. 942-948,
- [5] De Vleeschouwer, C., Delaigle, J. F., and Macq, B., 2003, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Trans. Multimedia*, 5(1), pp. 97--105.
- [6] Ni, Z., Shi, Y.Q, 2008, "Robust lossless image data hiding designed for semi-fragile image authentication." *IEEE Transactions on Circuits and Systems for Video Technology* 18(4), pp. 497-509.
- [7] Zeng, X., Ping, L., X. Pan, 2010, "A lossless robust data hiding scheme." *pattern recognition*, 43(4), pp. 1656-1667.
- [8] Yang, C.Y., and Lin, C.H., 2012, "High-Quality and Robust reversible data hiding by coefficient shifting algorithm," *ETRI Journal*, 34(3), pp. 429-438.
- [9] Thabit, R., and Khoo, B., 2015, "A new robust lossless data hiding scheme and its application to color medical images," *Digital Signal Processing*, 38, pp. 77-94.
- [10] Makbol, N.M., and B.E. Khoo, 2014, "A new robust and secure digital image watermarking scheme based on the integer wavelet transform and singular value decomposition." *Digital Signal Processing*, 33, pp. 134-147.
- [11] Chang, C.C., Lin, C.C., Tseng, C.S., and W.L. Tai, 2007, "Reversible hiding in DCT-based compressed images," *Information Sciences*, 177(13), pp. 2768-2786.
- [12] G. Xuan, Y.Q. Shin and Z. Ni, "Reversible data hiding for JPEG images based on histogram pairs," *Proc. of ICIAR 2007, LNCS 4633, PP.715-727, Springer-Verlag, 2007*
- [13] Wang, K., Lu, Z. M., and Hu, Y. J., 2013, "A high capacity lossless data hiding scheme for JPEG images," *Journal of Systems and Software*, 86(7), pp. 1965-1975.
- [14] Huang, F., X. Qu., Kim, H.J., and Huang, J., 2016, "Reversible Data Hiding in JPEG Images," *IEEE Transactions on Circuits and Systems for Video Technology*, 26(9), pp. 1610-1621.
- [15] JPEG recompression <http://ekot.dk/misc/recompress2/> Accessed 14 Dec 2016.