An Efficient Method for Imperceptive Data Embedding in Images[†]

Da-Chun Wu[1] and Wen-Hsiang Tsai[2]
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 300, Republic of China
e-mail: dcwu@mcu.edu.tw, whtsai@cis.nctu.edu.tw

Abstract

A novel and efficient method for embedding any form of secret messages into a gray-valued cover image is proposed. In the embedding process, a difference value is calculated from every non-overlapping pixel pair of the cover image. All possible difference values are quantized into a number of ranges. The selection of the range intervals is based on the characteristics of human visual system's sensitivity to gray value variation from smoothness to contrastiveness. The difference value then is replaced by a new value to embed the value of a sub-stream of the secret message. The number of bits which can be embedded in the pixel pair is decided by the width of the range that the difference value belongs to. The method was designed in such a way that the modification is never out of the range interval. This method provides an easy way to produce a more imperceptive result than those yielded by simple least-significant-bit replacement methods. The embedded secret message can be extracted out from the resulting stego-image without referencing the original cover image. Moreover, a pseudo-random mechanism may be used to achieve cryptography. Experimental results show the feasibility of the proposed method.

1. Introduction

Text, image, audio, and video can be represented as digital data. The explosion of Internet applications leads people into the digital world and communication via digital data becomes frequent. However, new issues also arose and have been explored [1], such as data security in digital communications, copyright protection of digitized properties, invisible communication via digital media, etc.

Data embedding in images is one way of information hiding which provides data security in digital media. It is an application in which secret messages are embedded in digital images. The secret message may be a serial number, a copyright logo, an annotation, a covert message, etc. Some terms about information hiding found in [2] are followed in this study. An image that holds a secret message is called a cover image, and the output of the hiding process, which includes the secret message, is called a stego-image. The secret message may be of any form of bit stream data, such as plain text, image, encrypted information, etc. Many techniques about embedding data in images have been proposed. A number of data embedding techniques are based on the method of replacing the least significant bits (LSB's) of the pixels of the cover image [3] and a pseudo-random number generation mechanism is often used to accomplish the security work [4]. In [5], an LSB-like embedding technique is used in a wavelet-based method by adding or subtracting one unit from the transforming coefficients of the image. In general, LSB embedding methods provide easy ways to embed large amounts of data in images with less perception but they are not robust with respect to some image processing procedures, such as lossy compression, noise adding, affine transformation, cropping, etc.

Some other methods based on modifying small details in images are also published in literature. A series of text marking methods [6] [7] [8] embed data by slightly shifting the contents in an electronic document. The texture block coding method [9] copies a small block with random texture into a region with similar texture. The patchwork method [10] changes the gray values of pixels by adding a value to the gray values of one set of pixels while subtracting the same value from another set. The fractal-based steganography method [11] creates a new range block, which is visually like the original range block, by transforming the selected domain block. These methods use the characteristic of the human visual system's low sensitivity to small changes in the gray values of small regions in images to embed data in the cover image. Image hiding can also be applied in frequency or other transform domains. Two of such methods are the use of randomly sequenced pulse position modulated codes [12], and the secure spread spectrum method [13].

[1] Also with the Department of Information Management, Ming Chuan University, Taipei, Taiwan 111, Republic of China.

[2] Corresponding author.

[†] This work is supported partially by National Science Council, R. O. C. under Grant No. NSC86-2213-E009-113.

Some data hiding techniques exploit the characteristics of the human visual system to guarantee that the modification made to the cover image is imperceptible. These methods are based on a model of frequency masking [14] or spatial masking [15] to calculate the visual threshold or the tolerable error level to embed data with imperception. Two of such techniques are [16] and [17]. These methods need more complex computations than LSB methods, but they are more robust.

In this paper, we propose a novel and efficient method to embed data in gray-valued images imperceptively. It was based on an LSB-like approach and a simple visual effect of the human visual system. We use the differences of the gray values in the two-pixel blocks of the cover image as features to cluster the blocks into a number of categories of smoothness and contrastiveness. Different amounts of data can be embedded in different categories according to the degree of smoothness or contrastiveness. This method provides an easy way to produce a more imperceptive result than those yielded by simple LSB replacement methods. The method was designed in such a way that there is no need of using the original image in recovering the secret message from the stego-image. Moreover, while hiding data we walk through the cover image in an order provided by a pseudo-random number generator to achieve cryptography, and so can prevent tampering access to the embedded data from illicit users.

The remainder of this paper is organized as follows. In Section 2, the proposed data embedding method is presented. The process for extracting the embedded data is described in Section 3. And several experimental results are illustrated in Section 4. Finally, concluding remarks as well as some suggestions for future works are stated in Section 5. In the appendix, the proof of an equation used in the data embedding process is included.

2. Proposed Data Embedding

Hiding data in the LSB's of the pixels of a gray-valued image is a common information hiding method that utilizes the characteristic of the human visual system's insensitivity to small changes in the image. This simple LSB embedding approach is easy for computation, and a large amount of data can be embedded with imperception. The more LSB's are used for embedding, the more distorted result will be produced. Not all pixels in an image can tolerate equal amounts of changes without causing notice the observer. The largest number of LSB's whose gray values can be changed without producing a perceptible result in each pixel is different. Changes of the gray values of pixels in smooth areas in images are more easily noticed by the human vision system. In the embedding method we propose, we simply divide the cover image into a number of non-overlapping two-pixel blocks. Each block is categorized according to the difference of the gray values of the two pixels in the block. A small difference value indicates that the block is in a smooth area and a large one indicates that it is in an edged area. The pixels in edged areas may, as mentioned previously, tolerate larger value changes than those in the smooth areas when the changes are judged with the same sensitivity by human perception. So, in the proposed method we embed more data in the edged areas and less data in the smooth areas. And it is in this way that we keep the resulting stego-image without noticeable changes.

A flowchart of the proposed embedding method is sketched in Fig. 1. The process of quantization of the differences of the gray values of two-pixel blocks and the process of data embedding are described subsequently.

2.1. Quantization of Differences of Gray Values of Two-Pixel Blocks

Cover images used in the proposed method are 256 gray-valued ones. We obtain a difference value d from every non-overlapping two-pixel block, say p_i and p_{i+1}, of a given cover image. Assume that the corresponding gray values of p_i and p_{i+1} are g_i and g_{i+1}, then d is computed as $g_{i+1} - g_i$, which may be in the range from -255 to 255. A block with d close to 0 is considered to be an extremely smooth block, which a block with d close to -255 or 255 is considered as a sharply edged block. By symmetry, we only consider the possible absolute values of d (0 through 255) and quantize them into a number of ranges, say R_i where $i = 1, 2, ..., n$. These ranges are assigned indices 1 though *n*. The lower bound and the upper bound values of R_i are denoted by l_i and u_i, respectively, where l_1 is 0 and u_n is 255. The width of R_i is $u_i - l_i + 1$. In our proposed method, each range width is taken to be in power of 2. This restriction of widths facilitates embedding binary data using the ranges. The range intervals we choose are based on the human visual system's capability mentioned previously. The widths of the ranges which represent the difference values of smooth blocks are chosen to be smaller while those which represent edged blocks are chosen to be larger. That is, we create ranges with smaller widths near 0 and ones with larger widths far away from 0 for the purpose of yielding better imperceptive results. A difference value which falls in a range with index *k* is said to have index *k*. All the values in a certain range (i.e., all the values with an identical index) are considered as close enough, and if a difference value in the range is replaced by another in the same range, the change presumably cannot be easily noticed by human perception. In the propose method, we embed some bits of the secret message into a two-pixel block by replacing the difference value of the block with one with an identical index, i. e., we change a difference value in one range into any of the difference value in the same range. In other words, in the proposed data embedding process, we adjust the gray values in each two-pixel pair by two new ones whose difference value causes changes unnoticeable to the observer of the stego-image. More details are described next.

2.2. Data Embedding

We consider the secret message as a long bit stream. We want to embed every bit in the bit stream into the two-pixel blocks of the cover image. The number of bits which can be embedded in each block varies and is decided by the width of the range to which the difference value of the two pixels in the block belongs. Given a two-pixel block *B* with index *k* and gray value difference *d*, the number of bits, say *n*, which can be embedded in this block, is calculated by $n = \log_2(u_k - l_k + 1)$. Since the width of each range is selected to be in power of 2, the value of $\log_2(u_k - l_k + 1)$ is an integer. A sub-stream *S* with *n* bits is selected next from the secret message for embedding in *B*. A new difference d' then is computed by

$$d' = \begin{cases} l_k + b & for\ d \geq 0; \\ -(l_k + b) & for\ d < 0, \end{cases} \quad (1)$$

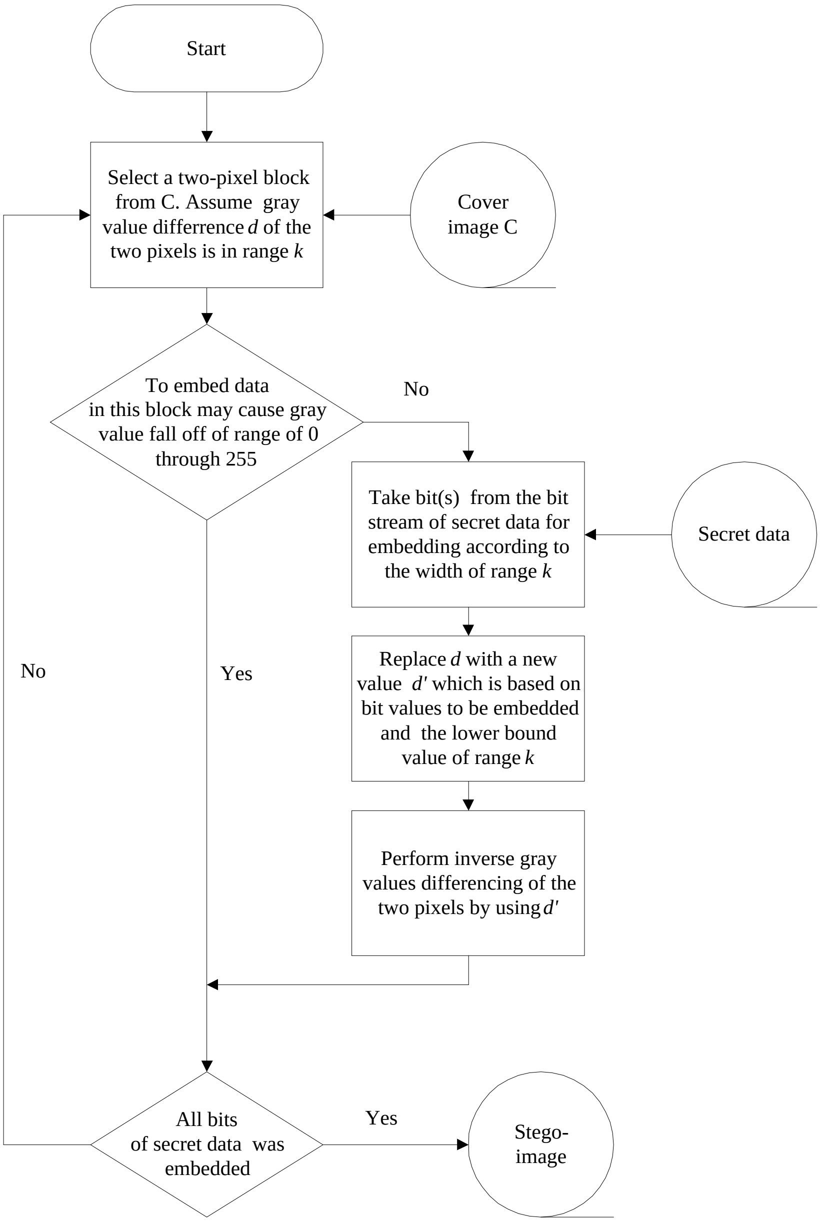

Start
Select a two-pixel block from C. Assume gray value differrence d of the two pixels is in range k
Cover image C
To embed data in this block may cause gray value fall off of range of 0 through 255
No
Take bit(s) from the bit stream of secret data for embedding according to the width of range k
Secret data
No
Yes
Replace d with a new value d' which is based on bit values to be embedded and the lower bound value of range k
Perform inverse gray values differencing of the two pixels by using d'
All bits of secret data was embedded
Yes
Stego-image

Figure 1: The data embedding process.

where b is the value of the sub-stream *S*. Because the value b is in the range from 0 to $u_k - l_k$, the value d' is in the range from l_k to u_k. According to previous discussions, if we replace d with d', the resulting changes are presumably unnoticeable to the observer. We then embed b by performing an inverse calculation from d' described next to yield the new gray values (g'_i, g'_{i+1}) for the pixels in the corresponding two-pixel block (p'_i, p'_{i+1}) of the stego-image. The embedding process is finished when all bits of the secret message are embedded.

The inverse calculation for computing (g'_i, g'_{i+1}) is based on a function $f((g_i, g_{i+1}), m)$ which is defined to be

$$
\begin{aligned}
& f((g_i, g_{i+1}), m) \\
&= (g'_i, g'_{i+1}) \\
&= \begin{cases} (g_i - ceiling_m,\ g_{i+1} + floor_m) \\ \qquad \text{if } d = g_{i+1} - g_i \text{ is an odd number;} \\ (g_i - floor_m,\ g_{i+1} + ceiling_m) \\ \qquad \text{if } d = g_{i+1} - g_i \text{ is an even number,} \end{cases} \qquad (2)
\end{aligned}
$$

where $m = d' - d$, $ceiling_m = \left\lceil \frac{m}{2} \right\rceil$, and $floor_m = \left\lfloor \frac{m}{2} \right\rfloor$. The above equation satisfies the requirement that the difference between g'_i and g'_{i+1} is d'. It is noted that a distortion reduction policy has been employed in designing Equation (2) for producing g'_i and g'_{i+1} from g_i and g_{i+1} so that the distortion caused by changing g_i and g_{i+1} is nearly equally distributed over the two pixels in the block. The effect is that the resulting gray value change in the block is less perceptible.

In the above inverse calculation, a smaller value of d' produces a smaller range interval between g'_i and g'_{i+1} while a larger d' produces a larger interval. So, (g_i, g_{i+1}) may produce invalid (g'_i, g'_{i+1}), i. e., some of the calculations may cause the resulting g'_i or g'_{i+1} to fall off the boundaries of the range [0, 255]. Although we may re-adjust the two new values into the valid range of [0, 255] by forcing a falling-off-boundary value to be one of the boundary values of 0 and 255, and adjusting the other to a proper value to satisfy the difference d', yet this might produce abnormal spots in contrast with the surrounding region in some cases. To solve this problem, we employ a checking process to detect such falling-off-boundary cases, and abandon the pixel blocks which yield such cases for data embedding. The gray values of the abandoned blocks are left without changes in the stego-image. This strategy helps us to distinguish easily blocks with embedded data from abandoned blocks in the process of recovering data from the stego-images, which will be discussed in the next section. It is noted that such abandoned pixel blocks are very few in real applications according to our experiments.

The proposed falling-off-boundary checking is proceeded by producing a pair of (g'_i, g'_{i+1}) from the inverse calculation of the value of the function $f((g_i, g_{i+1}), u_k - d)$. Since u_k is the maximum value in the range from l_k to u_k, the resulting pair of (g'_i, g'_{i+1}) produced by the use of u_k will yield the maximum range interval $g'_{i+1} - g'_i$, compared with those yielded by the (g'_i, g'_{i+1}) pairs that are produced by the use of other values of d' in the range from l_k to u_k. That is, this maximum range interval $g'_{i+1} - g'_i$ covers all of the ranges yielded by the other (g'_i, g'_{i+1}) pairs. So, the falling-off-boundary checking for the block can be proceeded by only examining the values of (g'_i, g'_{i+1}) which are produced by the case of using u_k. If either g'_{i+1} or g'_i falls off the boundary of 0 or 255, we regard the block to have the possibility of falling off, and abandon the block for embedding data.

The inverse calculation in Eq. (2) is designed in such a way that it satisfies the following property:

$$
f((g_i, g_{i+1}), m) = f(f((g_i, g_{i+1}), m'), m'') \quad \text{for } m = m' + m''. \qquad (3)
$$

The proof can be found in the appendix. This equation means that the inverse calculation can be proceeded directly or progressively. This property is useful for judging the existence of embedded data in each block in the data recovering process.

An illustration of the data embedding process is shown in Fig. 2. In the figure, the gray values of a sample two-pixel block are assumed to be (50, 65). The difference value is 15, which is in the range of 8 through 23. The width of the range is $16 = 2^4$, which means that a difference value in the range can be used to embed four bits of secret data. Assume that the four leading bits of the secret data are 1010. The value of this bit stream is 10. It is added to the lower bound value 8 of the range to yield the new difference value 18. Finally, by Eq. (2) the values (48, 66) are computed for use as the gray values in the stego-image. Note that $66 - 48 = 18$.

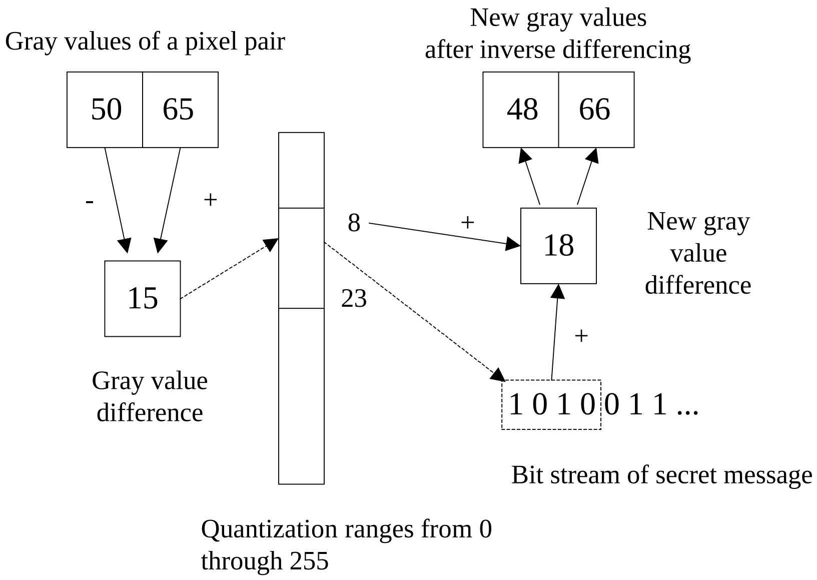

Gray values of a pixel pair
50
65
-
+
15
Gray value difference
Quantization ranges from 0 through 255
8
23
+
18
New gray value difference
New gray values after inverse differencing
48
66
+
1 0 1 0 0 1 1 ...
Bit stream of secret message

Figure 2: An illustration of the data embedding process.

3. Process of Recovering Embedded Data from Stego-Images

The process of extracting the embedded message is proceeded by using the seed of the pseudo-random scheme to produce the same traversing order for visiting the two-pixel blocks as in the embedding process. Each time we visit a two-pixel block in the stego-image, we apply the same falling-off-boundary checking as mentioned previously to the block to find out whether the block was used or not in the embedding process. Assume that the block in the stego-image has the gray values (g_i^*, g_{i+1}^*), and that the difference d^* of the two gray values is with index *k*. We apply the falling-off-boundary checking process to (g_i^*, g_{i+1}^*) by using $f((g_i^*, g_{i+1}^*), u_k - d^*)$.

We now want to prove that the resulting (g_i'', g_{i+1}'') computed from $f((g_i^*, g_{i+1}^*), u_k - d^*)$ are identical to the gray values (g_i', g_{i+1}') which were computed by $f((g_i, g_{i+1}), u_k - d)$ in the embedding process. The proof is as follows. First,

$$\begin{aligned}&(g_i', g_{i+1}')\\ &= f((g_i, g_{i+1}), u_k - d)\\ &= f((g_i, g_{i+1}), d^* - d + u_k - d^*).\end{aligned} \tag{4}$$

By Eq. (3), the above result can be transformed further to be:

$$\begin{aligned}&f((g_i, g_{i+1}), d^* - d + u_k - d^*)\\ &= f(f((g_i, g_{i+1}), d^* - d), u_k - d^*)\\ &= f((g_i^*, g_{i+1}^*), u_k - d^*)\\ &= (g_i'', g_{i+1}'').\end{aligned} \tag{5}$$

This completes the proof.

The above property shows that the results of both of the falling-off-boundary checking processes, one in data embedding and the other in data recovery, are identical. This in turn implies that if either of the gray values of the computed values (g_i'', g_{i+1}'') falls off the boundaries of the range [0, 255], it means that the current block was not used for embedding data, or that the block was abandoned in the embedding process. On the contrary, if both of the values (g_i'', g_{i+1}'') do not fall off the range, it means that some data was embedded in the block. The value *b*, which was embedded in this two-pixel block, is then extracted out using the equation

$$b = \begin{cases} d^* - l_k & \text{for } d^* \geq 0; \\ -d^* - l_k & \text{for } d^* < 0. \end{cases} \tag{6}$$

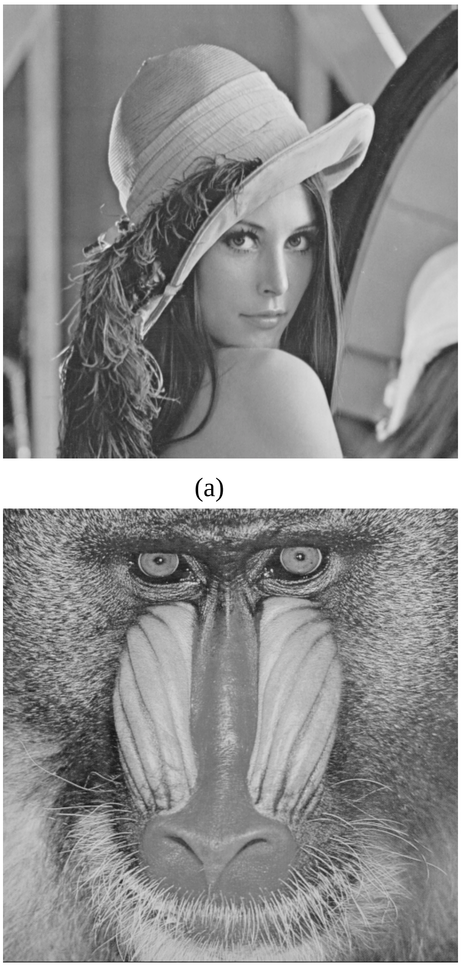
Note that in the recovery of the secret message from the stego-image using the previously-described extraction process, there is no need of referencing the cover image.

4. Experimental Results

In our experiments, four cover images "Lena", "Jet", "Peppers", and "Baboon" were used, each with size 512×512. Two of them are shown in Fig. 3. We used a Word-formatted file (50176 bytes) which consists of the text of this article as the secret message in the experiments. Two sets of widths of ranges of gray value differences were used in the simulations. The first simulation were based on selecting the range widths of 8, 8, 16, 32, 64, and 128, which partition the total range of [0, 255] into [0, 7], [8, 15], [16, 31], …, [128, 255]. Two of the stego-images resulting from embedding part of the secret data using such a set of range widths are shown in Fig. 4. The second simulation was based on the range widths of 32, 64, 128, and 32. Two of the resulting stego-images are shown in Fig. 5. All of the results were produced by embedding the secret data in the two-pixel blocks of the cover image in a random traversing order generated by a pseudo-random scheme, which walks through the cover image and visits each two-pixel block only once. It is thus shown that the proposed hiding method can embed data without noticeable changes. The values of the peaks of the signal-to-noise (PSNR) and the root-mean-square error (RMSE) of the embedding results are shown in Table 1. The quality is still good even in a stego-image with a lower PSNR.

5. Conclusion

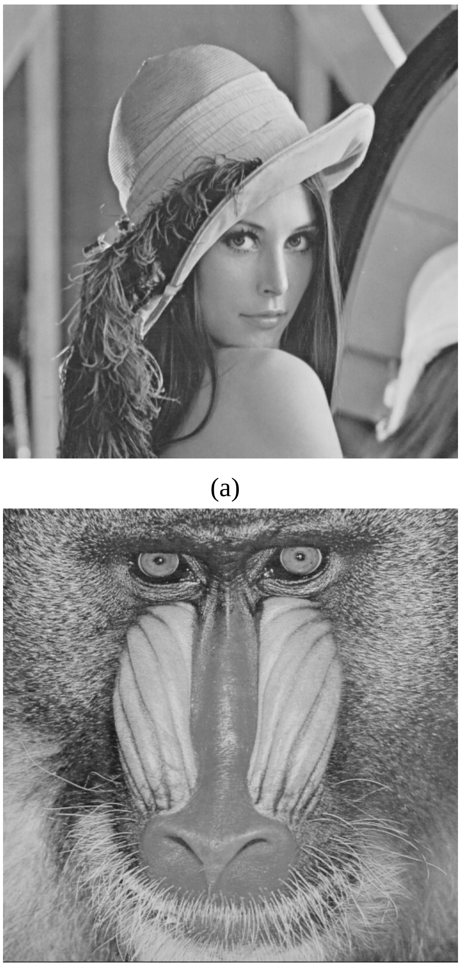
A novel method for embedding any kind of secret message into an image without producing noticeable changes has been proposed. There is no need of referencing the original image when extracting out the embedded data from a stego-image. The method utilizes the characteristic of the human visual system’s sensitivity to gray value variation. The method embeds secret data by replacing the difference values of the two-pixel blocks of the cover image with similar ones in which bits of embedded data are included. The method not only provides a better way for embedding large amounts of data into



(a)

(b)

Figure 3: The cover images (512×512). (a) "Lena", and (b) "Baboon".

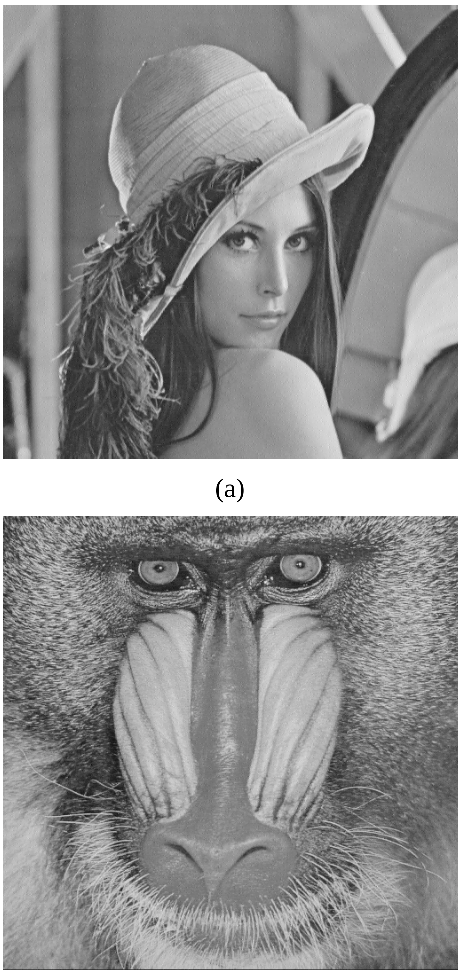


(a)

(b)

Figure 4: Two of the resulting images after embedding a Word-format file consisting of the text of this article by using a set of range widths of 8, 8, 16, 32, 64, and 128.

cover images with imperception, but also offers an easy way to accomplish cryptography. This embedding method can be extended easily to embed data in other digital media such as audio and video.


(a)

(b)

Figure 5: Two of the resulting images after embedding a Word-format file consisting of the text of this article by using a set of range widths of 32, 64, 128, and 32.

Table 1
Values of RMSE's and PSNR's of stego-images that embed a file consisting of the text of this article by using two sets of range widths in the embedding process.

Cover image	Embedding by using the range widths of 8, 8, 16, 32, 64, and 128		Embedding by using the range widths of 32, 64, 128, and 32	
	RMSE	PSNR	RMSE	PSNR
Lena	2.14	41.54	4.41	35.23
Jet	2.30	40.90	4.55	34.97
Peppers	2.24	41.13	4.40	35.25
Baboon	3.17	38.11	5.56	33.22

References

[1] J. Zhao, E. Koch, and C. Luo, "Digital Watermarking In business Today and Tomorrow," *Communication of the ACM*, vol. 41, No. 7, pp. 67-74, 1998.

[2] B. Pfitzmann, "Information Hiding Terminology," *Proc. First Int'l Workshop Information Hiding, Lecture Notes in Computer Science* No. 1174, Springer-Verlag, Berlin, 1996, pp. 347-349.

[3] S. Walton, "Image Authentication for a Slippery New Age," *Dr. Dobb's Journal: Software Tools For The Professional Programmer*, vol. 20, no. 4, pp. 18-26, April 1995.

[4] L. F. Turner, "*Digital Data Security System*," Patent IPN, WO 89/08915, 1989.

[5] J. Ohnishi and K. Matsui, "Embedding A Seal Into A Picture Under Orthogonal Wavelet Transform," in *Proceedings of Multimedia'96*. Piscataway, NJ: IEEE Press, 1996, pp. 514-521.

[6] N. F. Maxechuk, "Electronic Document Distribution," *AT&T Technical Journal*, vol. 73, no. 5, pp. 73-80, 1994.

[7] J. T. Brassil, S. Low, N. F. Maxemchuk, and L. O'Gorman, "Electronic Marking and Identification Technique to Discourage Document Copying," *IEEE J. Selected Areas Commum*., vol. 13, no. 8, pp.1495-1503, 1995.

[8] S. H. Low and N. F. Maxemchuk, "Performance Comparison of Two Text Marking Methods," *IEEE Journal on Selected Areas in Communication*, vol. 16, No. 4, pp. 561-572, 1998.

[9] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding," *IBM System Journal*, vol. 35(3/4), pp. 313-336, 1996.

[10] W. Bender, N. Morimoto, and D. Gruhl, "Method and Apparatus for Data Hiding in Images," U. S. Patent, No. 5689587, 1997.

[11] P. Davern and M. Scott "Fractal Based Image Steganography," *Proc. First Int'l Workshop Information Hiding, Lecture Notes in Computer Science* No. 1174, Springer-Verlag, Berlin, 1996, pp. 279-294.

[12] E. Koch and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling," *Proc. IEEE Nonlinear Signal and Image Processing Workshop*, Thessaloniki, Greece, 1995, pp. 452-455.

[13] I. J. Cox, J. Kilian, F. T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687, 1997.

[14] G. Legge and J. Foley, "Contrast masking in human vision," *J. Opt. Soc. Amer.*, vol. 70, no. 12, pp. 1458-71, 1990.

[15] B. Girod, "The information theoretical significance of spatial and temporal masking in video signals," *Porc. SPIE Human Vision, Visual Processing, and Digital Display*, 1989, vol. 1077, pp. 178-187.

[16] M. D. Swanson, B. Zin, and A. H. Tewfik, "Multiresolution Scene-Based Video Watermarking Using Perceptual Models," *IEEE Journal on Selected Areas in Communication*, vol. 16, No. 4, pp. 540-550, 1998.

[17] C. I. Podilchuk, and W. Zeng, "Image-Adaptive Watermarking Using Visual Models," *IEEE Journal on Selected Areas in Communication*, vol. 16, No. 4, pp. 525-539, 1998.

Appendix: Proof of Equation (3)

Another way to represent Eq. (2) is as follows:

$$(g_i', g_{i+1}') = \begin{cases} (g_i - \frac{m+1}{2}, g_{i+1} + \frac{m-1}{2}) & \\ \quad \text{when } g_{i+1} - g_i \text{ is odd and } m \text{ is odd;} & (7) \\ (g_i - \frac{m-1}{2}, g_{i+1} + \frac{m+1}{2}) & \\ \quad \text{when } g_{i+1} - g_i \text{ is even and } m \text{ is odd;} & (8) \\ (g_i - \frac{m}{2}, g_{i+1} + \frac{m}{2}) & \\ \quad \text{when } m \text{ is even,} & (9) \end{cases}$$

where $m = d' - d$ represents the total changes of the gray values of g_i and g_{i+1} to produce g_i' and g_{i+1}'. It is easy to verify that the range interval of the resulting pair of (g_i', g_{i+1}') produced by a positive m will cover that of (g_i, g_{i+1}), that is, the produced range will be enlarged. On the contrary, a negative m will produce a reduced range. In the following, we do not use Eq. (2) but use Eq. (7) through (9) instead.

The proof of the correctness of Eq. (3) is proceeded by considering different combinations of m, m', and m'' which meet the condition of $m = m' + m''$. Possible combinations include:

(I) m is even, m' is odd, and m'' is odd;

(II) m is even, m' is even, and m'' is even;

(III) m is odd, m' is odd, and m'' is even;

(IV) m is odd, m' is even, and m'' is odd.

The proof of Case (I) is conducted in the following by considering two possible situations, namely, when the value $g_{i+1} - g_i$ is even and when it is odd.

Firstly, if $g_{i+1} - g_i$ is even, then by Eq. (8) we have $f((g_i, g_{i+1}), m') = (g_i - \frac{m'-1}{2}, g_{i+1} - \frac{m'+1}{2})$. It is easy to see that the difference between $g_i - \frac{m'-1}{2}$ and $g_{i+1} + \frac{m'+1}{2}$ is odd. Therefore,

$$\begin{aligned} & f(f((g_i, g_{i+1}), m'), m'') \\ &= f((g_i - \frac{m'-1}{2}, g_{i+1} + \frac{m'+1}{2}), m'') && \text{By Eq. (8)} \\ &= (g_i - \frac{m'-1}{2} - \frac{m''+1}{2}, g_{i+1} + \frac{m'+1}{2} + \frac{m''-1}{2}) \\ & && \text{By Eq. (7)} \\ &= (g_i - \frac{(m'+m'')}{2}, g_{i+1} + \frac{(m'+m'')}{2}) \\ &= f((g_i, g_{i+1}), m'+m'') && \text{By Eq. (9)} \\ &= f((g_i, g_{i+1}), m) \end{aligned}$$

which is just the desired Eq. (3).

Secondly, if $g_{i+1} - g_i$ is odd, then by Eq. (7) we have $f((g_i, g_{i+1}), m') = (g_i - \frac{m'+1}{2}, g_{i+1} - \frac{m'-1}{2})$. It is easy to see that the difference between $g_i - \frac{m'+1}{2}$ and $g_{i+1} + \frac{m'-1}{2}$ is even. Therefore,

$$\begin{aligned} & f(f((g_i, g_{i+1}), m'), m'') \\ &= f((g_i - \frac{m'+1}{2}, g_{i+1} + \frac{m'-1}{2}), m'') && \text{By Eq. (7)} \\ &= (g_i - \frac{m'+1}{2} - \frac{m''-1}{2}, g_{i+1} + \frac{m'-1}{2} + \frac{m''+1}{2}) \\ & && \text{By Eq. (8)} \\ &= (g_i - \frac{(m'+m'')}{2}, g_{i+1} + \frac{(m'+m'')}{2}) \\ &= f((g_i, g_{i+1}), m'+m'') && \text{By Eq. (9)} \\ &= f((g_i, g_{i+1}), m) \end{aligned}$$

which is just the desired Eq. (3).

The proofs of the other cases can be proceeded similarly, and are omitted.