Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Pattern Recognition Letters 24 (2003) 1613–1626

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A steganographic method for images by pixel-value differencing

Da-Chun Wu [a], Wen-Hsiang Tsai [b,*]

[a] *Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung 811, Taiwan, ROC*

[b] *Department of Computer and Information Science, National Chiao Tung University, 1001 Ta Hsueh Rd., Hsinchu 30050, Taiwan, ROC*

Received 29 October 2001; received in revised form 11 December 2002

Abstract

A new and efficient steganographic method for embedding secret messages into a gray-valued cover image is proposed. In the process of embedding a secret message, a cover image is partitioned into non-overlapping blocks of two consecutive pixels. A difference value is calculated from the values of the two pixels in each block. All possible difference values are classified into a number of ranges. The selection of the range intervals is based on the characteristics of human vision's sensitivity to gray value variations from smoothness to contrast. 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 a pixel pair is decided by the width of the range that the difference value belongs to. The method is 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 imperceptible result than those yielded by simple least-significant-bit replacement methods. The embedded secret message can be extracted from the resulting stego-image without referencing the original cover image. Moreover, a pseudorandom mechanism may be used to achieve secrecy protection. Experimental results show the feasibility of the proposed method. Dual statistics attacks were also conducted to collect related data to show the security of the method.

Keywords: Steganography; Data hiding; Cover image; Stego-image; Security

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 (Artz, 2001; Zhao et al., 1998), such as data security in digital communications, copyright protection of digitized properties, invisible communication via digital media, etc.

The term steganography is derived from the Greek language and means covert writing. It is the technique of encoding secret information in a

[*] Corresponding author. Tel.: +886-35-720631/712121; fax: +886-35-727382/721490.

E-mail addresses: dcwu@ccms.nkfust.edu.tw (D.-C. Wu), whtsai@cis.nctu.edu.tw (W.-H. Tsai).

0167-8655/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0167-8655(02)00402-6

communication channel in such a manner that the very existence of the information is concealed. Computer-based image steganography (Anderson and Petitcolas, 1998) is one way of data hiding which provides data security in digital images. It is considered as a technique inspired from ancient steganography. The aim is to embed and deliver secret messages in digital images without any suspiciousness. The secret message might be a caption, a plain text, another image, a control signal, or anything that can be represented in bit stream form. The secret message may be compressed and encrypted before the embedding steps begin.

Another way of data hiding is image watermarking (Swanson et al., 1998; Hartung and Kutter, 1999; Voyatzis and Pitas, 1999; Wolfgang et al., 1999; Cox et al., 2001) which is a novel way for embedding watermark information in host images. The main purposes of image watermarking include copyright protection (Podilchuk and Zeng, 1998; Nikolaidis and Pitas, 1998; Koch and Zhao, 1995; Cox et al., 1997; Podilchuk and Delp, 2001) and authentication (Yeung and Mintzer, 1997; Fridrich, 1998). In such applications, it is required that the embedded information be unaltered, or altered only up to an acceptable degree of distortion, no matter how the watermarked image is attacked. Many other new applications of watermarking are also introduced (Cox et al., 2000), such as broadcast monitoring, proof of ownerships, transactional watermarks, copy control, and covert communication.

Computer-based image steganography mainly considers the requirement that the steganographic result, the so-called stego-image, be undetectable, as pointed out in (Anderson and Petitcolas, 1998). Such steganographic techniques may be used in various applications. In the application of image database retrieval, auxiliary information, like captions, time stamps, news, etc., may be embedded into images for convenience of simultaneous handling of the images and the embedded information. In such a kind of application, although the existence of the embedded data may be publicly known, the systems are basically closed and there is no worry about the possibility of being attacked from outside worlds. Non-robust data embedding methods are appropriate here. On the other hand, digital images may be used as the carriers of secret messages in steganographic methods to deliver or hide data like secret letters, military maps, favorite pictures, etc. In such data embedding applications with emphasis on cheating or hiding, attackers do not know that the stego-image has included secret messages, so they will not disturb the stego-image. And so non-robust techniques may also be used here. To explore a large amount of space for efficiently embedding data into an image is an emphasis of such steganographic studies.

Many steganographic 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 (LSBs) of the pixels of the cover image (Walton, 1995) and a pseudo-random number generation mechanism is often used to accomplish the security work (Turner, 1989). In (Ohnishi and Matsui, 1996), an LSB-like embedding technique is used in a wavelet-based method by adding or subtracting one unit from the transform coefficients of the image.

Some other steganographic methods based on modifying small details in images are also published in the literature. A series of text marking methods embedding data by slightly shifting the contents in an electronic document have been conducted by Maxemchuk (1994). The texture block coding method proposed in (Bender et al., 1996) copies a small block with random texture into a region with similar texture. The fractal-based steganography method proposed in (Davern and Scott, 1996) creates a new range block, which is visually like the original range block, by transforming the selected domain block. A method that uses color palettes for hiding data is proposed by Johnson and Jajodia (1998). Non-adaptive and adaptive steganographic techniques for images in palette format were reported by Fridrich and Rui (2000). Some more literature about computer-based image steganography can also be found (Anderson and Petitcolas, 1998; Franz and Pfitzmann, 2000; Petitcolas et al., 1999; Ramkumar and Akansu, 2001). Some steganalytic methods for detecting the presence of hidden information in images, such as visual attacks and statistical attacks, have also been explored in (Fridrich and

Goljan, 2002; Westfeld and Pfitzmann, 2000). The dual statistics methods proposed by Fridrich et al. (2001) can be employed to detect the existence of LSB steganography in images.

In this paper, we propose a new and efficient steganographic method to hide data in gray-valued images imperceptibly with no consideration of robustness. It was based on a simple visual effect of the human visual perception capability. 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 contrast properties. Different amounts of data can be embedded in different categories according to the degree of smoothness or contrast. This method provides an easy way to produce more imperceptible results 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 the cover image is traversed in an order provided by a pseudo-random number generator to achieve secrecy, and so to 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. 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 A, the proof of an equation used in the data embedding process is included.

2. Proposed data embedding

Hiding data in the LSBs of the pixels of a gray-valued image is a common information hiding method that utilizes the characteristic of the human vision'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 without great quality loss. The more LSBs 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 to an observer. The largest number of LSBs whose gray values can be changed without producing a perceptible artifact in each pixel is different. Changes of the gray values of pixels in smooth areas in images are more easily noticed by human eyes. 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 changes of pixel values than those in the smooth areas. So, in the proposed method we embed more data in edged areas than in the smooth areas. And it is in this way that we keep the changes in the resulting stego-image unnoticeable.

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

The cover images used in the proposed method are 256 gray-valued ones. A difference value d is computed from every non-overlapping block of two consecutive pixels, say p_i and p_{i+1}, of a given cover image. The way of partitioning the cover image into two-pixel blocks runs through all the rows of each image in a zigzag manner, as shown in Fig. 2. Assume that the gray values of p_i and p_{i+1} are g_i and g_{i+1}, respectively, 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, whereas 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 classify them into a number of contiguous ranges, say R_i where $i = 1, 2, \ldots, n$. These ranges are assigned indices 1 though n. The lower and upper bound values of R_i are denoted by l_i and u_i,

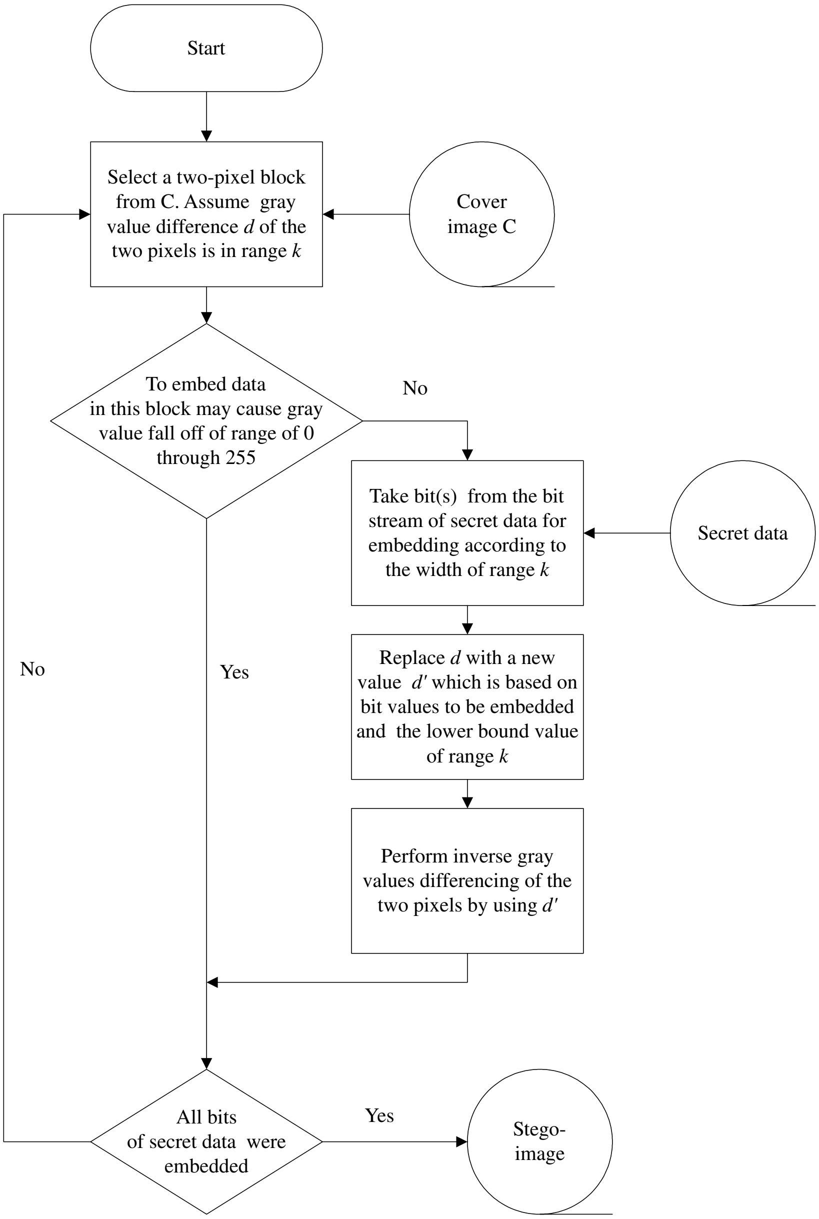
Start
Select a two-pixel block from C. Assume gray value difference *d* of the two pixels is in range *k*
Cover image C
To embed data in this block may cause gray value fall off of range of 0 through 255
No
Take bit(s) from the bit stream of secret data for embedding according to the width of range *k*
Secret data
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'*
No
Yes
All bits of secret data were embedded
Yes
Stego-image

Fig. 1. The data embedding process.

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, the width of each range is taken to be a power of 2. This restriction of widths facilitates embedding binary data. The selected range intervals are based on the human visual 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 the difference values of edged blocks are chosen to be larger. That is, we create ranges with smaller widths when d is close to 0 and ones with larger widths when d is far away from 0 for the purpose of yielding better imperceptible results. A differ-

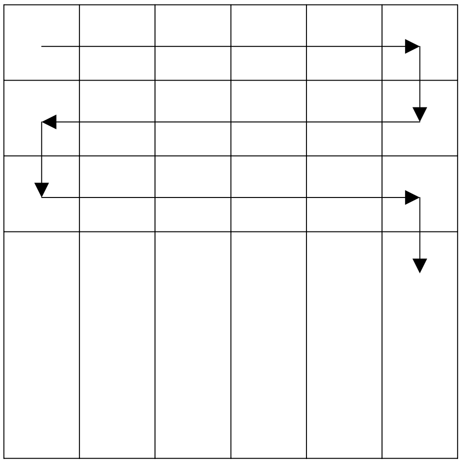

Fig. 2. The non-overlapping two-pixel blocks are constructed by grouping every two consecutive pixels in a cover image in a zigzag scanning of the image rows.

ence 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*. That is, if a difference value in a range is replaced by another in the same range, the change presumably cannot be easily noticed by human eyes. In the proposed 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 values 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 an 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 a power of 2, the value of $n = \log_2(u_k - l_k + 1)$ 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

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

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 of d' is in the range from l_k to u_k. According to the 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 the bits of the secret message are embedded.

The inverse calculation for computing (g'_i, g'_{i+1}) from the original gray values (g_i, g_{i+1}) of the pixel pair is based on a function $f((g_i, g_{i+1}), m)$ which is defined to be

$$f((g_i, g_{i+1}), m) = (g'_i, g'_{i+1})$$
$$= \begin{cases} (g_i - \text{ceiling}_m, g_{i+1} + \text{floor}_m) \\ \quad \text{if } d \text{ is an odd number;} \\ (g_i - \text{floor}_m, g_{i+1} + \text{ceiling}_m) \\ \quad \text{if } d \text{ is an even number,} \end{cases} \tag{2}$$

where $m = d' - d$, $\text{ceiling}_m = \lceil m/2 \rceil$, and $\text{floor}_m = \lfloor m/2 \rfloor$. 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 Eq. (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 intact 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 a stego-image, 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 proceeds by producing a pair of $(\hat{g}_i, \hat{g}_{i+1})$ 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 $(\hat{g}_i, \hat{g}_{i+1})$ produced by the use of u_k will yield the maximum difference. That is, this maximum range interval $\hat{g}_{i+1} - \hat{g}_i$ covers all of the ranges yielded by the other $(\hat{g}_i, \hat{g}_{i+1})$ pairs. So, the falling-off-boundary checking for the block can proceed by only examining the values of $(\hat{g}_i, \hat{g}_{i+1})$ which are produced by the case of using u_k. If either $\hat{g}_{i+1}$ or $\hat{g}_i$ 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.

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

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

The proof can be found in Appendix A. This equation means that the inverse calculation can proceed 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. 3. 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

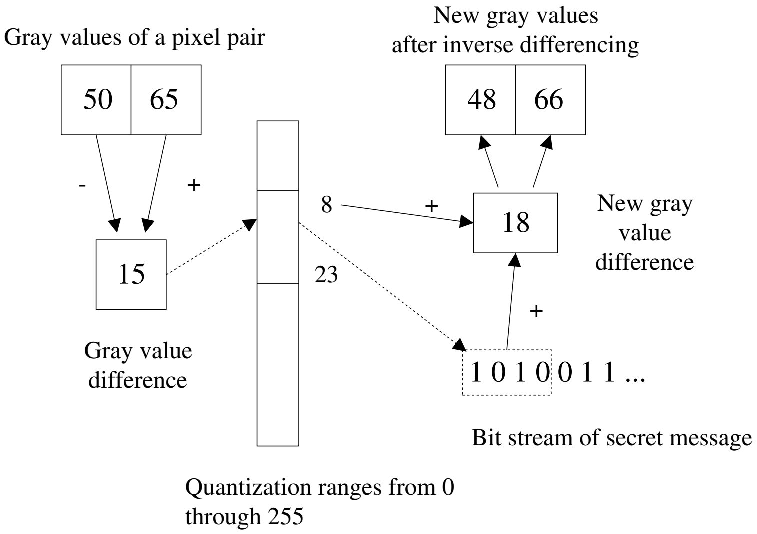
Gray values of a pixel pair
50
65
-
+
15
Gray value
difference
8
23
Quantization ranges from 0
through 255
New gray values
after inverse differencing
48
66
+
18
New gray
value
difference
+
1 0 1 0 0 1 1 ...
Bit stream of secret message

Fig. 3. An illustration of the data embedding process.

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 $66-48=18$.

3. Process of recovering embedded data from stego-images

The process of extracting the embedded message proceeds 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 $(\hat{g}_i^*, \hat{g}_{i+1}^*)$ computed from $f((g_i^*, g_{i+1}^*), u_k - d^*)$ are identical to the gray values $(\hat{g}_i, \hat{g}_{i+1})$ which were computed by $f((g_i, g_{i+1}), u_k - d)$ in the embedding process. The proof is as follows. First,

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

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

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

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 $(\hat{g}_i^*, \hat{g}_{i+1}^*)$ 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 $(\hat{g}_i^*, \hat{g}_{i+1}^*)$ 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

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

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.

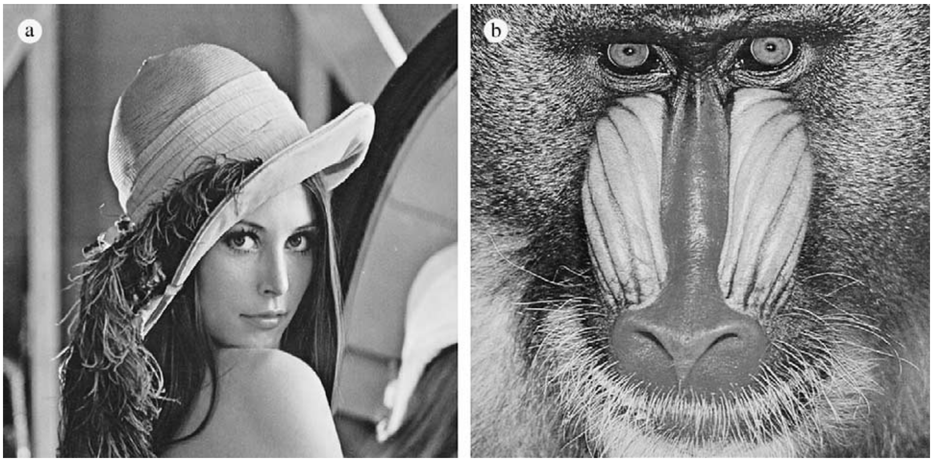

Fig. 4. The cover images (512 × 512) (a) “Lena”, and (b) “Baboon”.

4. Experimental results and discussions

4.1. 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. 4. Two sets of widths of ranges of gray value differences were used in the experiments. The first experiment was based on selecting the range widths of 8, 8, 16, 32, 64, and 128, which partition the total range of $[0, 255]$ into $[0, 7], [8, 15], [16, 31], \ldots, [128, 255]$. The second experiment was based on the use of the range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64. The values of the capacities for embedding data by using the cover image and the two sets of range widths are given in

Table 1
Values of the capacities for embedding data by using the cover image

Cover image	Maximum capacity	
	Embedding using the range widths of 8, 8, 16, 32, 64, and 128	Embedding using the range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64
Lena	50960	25940
Jet	51243	24177
Peppers	50685	27269
Baboon	56291	36061

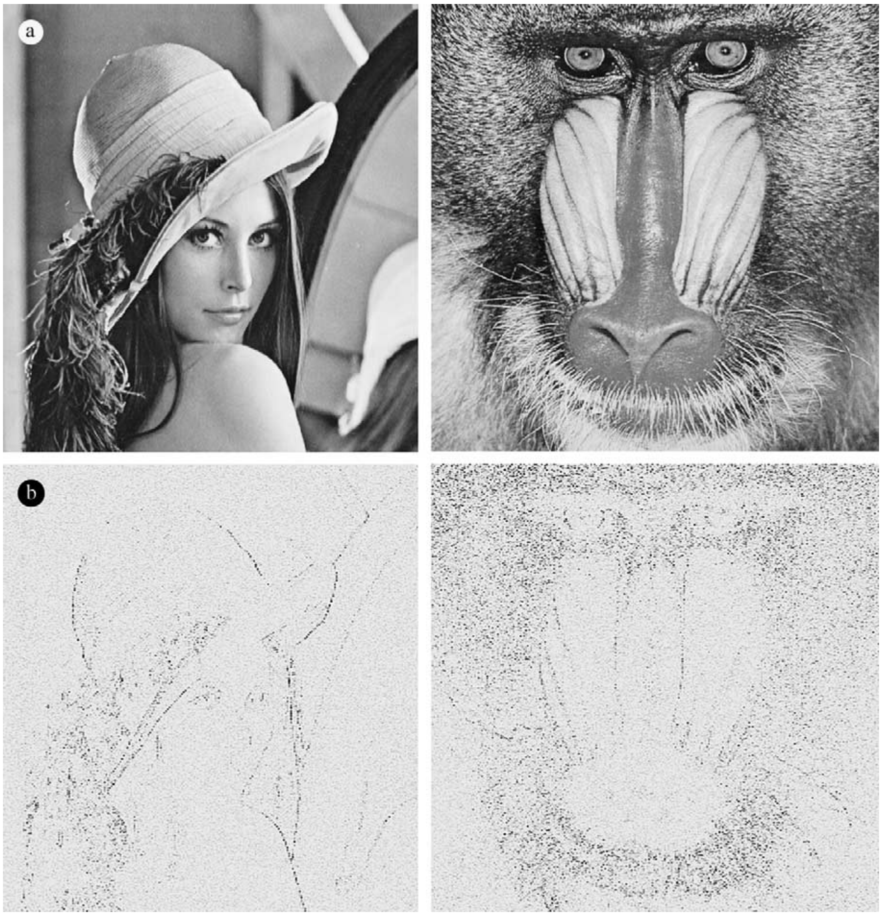

Fig. 5. Two of the resulting images and their enhanced difference images after embedding a Word-format file consisting of the text of this article by using a set of range widths of 8, 8, 16, 32, 64, and 128. (a) The stego-images, (b) the enhanced difference images between the cover images and the stego-images.

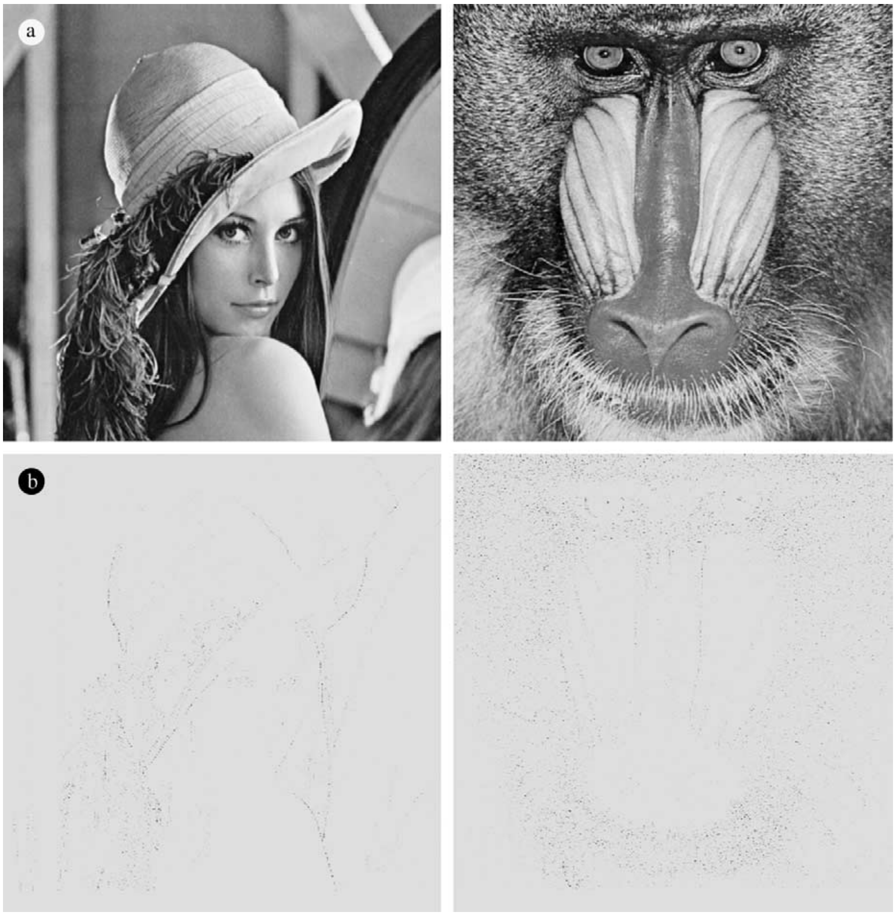

Fig. 6. Two of the resulting images and their enhanced difference images after embedding a Word-format file consisting of the text of this article by using a set of range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64. (a) The stego-images, (b) the enhanced difference images between the cover images and the stego-images.

Table 1. We used a Word-formatted file which consists of the text of this article as the secret message in the experiments. Some results of the first experiment are shown Fig. 5. In Fig. 5(a), we show the two of the stego-images resulting from embedding the given secret data using the first set of range widths, and in Fig. 5(b) we show the corresponding enhanced difference images between the stego-images of Fig. 5(a) and the cover images of Fig. 4 (with the differences of gray values being scaled five times). Similarly results of the second experiment are shown in Fig. 6. The difference images are shown here to indicate the distortions resulting from the data embedding process. From them, we see that most of the distortions are found on the edges in the images. This means that such distortions will be less noticeable because changes in edge parts of images are generally less obvious to human eyes.

In contrast, for the purpose of comparison, two stego-images resulting from embedding random data into the three LSBs of the pixel values using the conventional LSB-embedding steganographic technique, and the corresponding enhanced difference images are shown in Fig. 7. It is seen from this figure that distortions are spread *all over the image* which are more obvious to observers than distortions resulting from our method which are limited essentially *at edge areas*, as shown in Figs. 5 and 6.

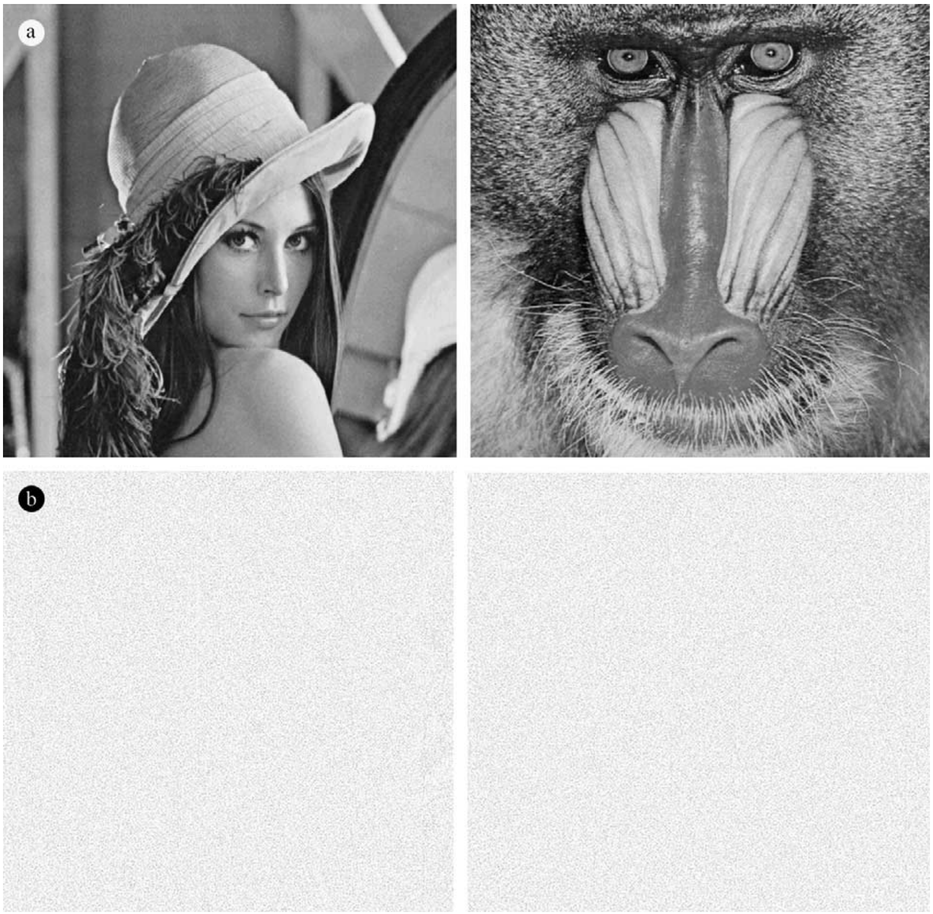

Fig. 7. Two of the resulting images and their enhanced difference images after embedding random data in the three LSBs. (a) The stego-images, (b) the enhanced difference images between the cover images and the stego-images.

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.

Finally, 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 2. The quality is still good even in a stego-image with a lower PSNR.

Table 2
Values of RMSEs and PSNRs of stego-images in which a file consisting of the text of this article is embedded using two sets of range widths in the embedding process

Cover image	Embedding using the range widths of 8, 8, 16, 32, 64, and 128		Embedding using the range widths of 2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, and 64	
	RMSE	PSNR	RMSE	PSNR
Lena	2.07	41.79	0.97	48.43
Jet	2.28	40.97	1.09	45.67
Peppers	2.09	41.73	1.20	47.19
Baboon	3.25	37.90	1.59	44.10

4.2. Discussions

The proposed method may be used to embed variable numbers of bits into blocks of two pixels. It does not replace the LSBs of pixel values directly; instead, it changes the differences of the two pixel values in a block. We cannot find obvious suspicious artifacts on the resulting images or bitplanes by simple visual inspection.

To check whether the function of the proposed embedding method can be detected with some newly announced related statistical steganalytic techniques, we tested the stego-images yielded by our method with the dual statistics method proposed by Fridrich et al. (2001) as a demonstration. By the method, an image is divided into disjoint pixel groups. The regularity of each group is computed by a discrimination function. By combining the function and an invertible operation, three types of pixel groups are defined: regular, singular, and unusable. Two complemental masks are used for simulating the act of different noise adding. Some test results using this method were transformed into the two diagrams shown in Fig. 8. In the diagrams, the x-axes depict the percentage of image pixels into which message data are embedded, and the y-axes depict the relative numbers (in percentage) of regular and singular pixel groups with masks $m = [0\,1\,1\,0]$ and $-m = [0\,-1\,-1\,0]$. Such diagrams are referred to as RS-diagrams. According to Fridrich et al. (2001), if more and more LSBs are replaced with random data, then the percentages R_m and S_m of the two pixel groups (regular and singular, respectively) in the diagram will become equal gradually when the mask of m is adopted in the statistics analysis process, or the percentages R_{-m} and S_{-m} will become more and more unequal when the mask of m is adopted. From the RS-diagram of Fig. 8(a), we can see that the function of conventional LSB-embedding steganographic techniques in images indeed can be detected because when the percentage of pixels embedded with data into their LSBs approaches 100%, the percentages of the regular and singular pixel groups will become more and more equal or unequal. But the results of our method as shown by the RS-diagram of Fig. 8(b) indicate that the stego-images seemingly do not contain any embedded data in their LSBs, because the expected value of R_m is seen close to that of R_{-m} and so are the case of S_m and S_{-m}, i.e., $R_m \cong R_{-m}$ and $S_m \cong S_{-m}$. This proves that our steganographic method is secure from the viewpoint of the dual statistics method.

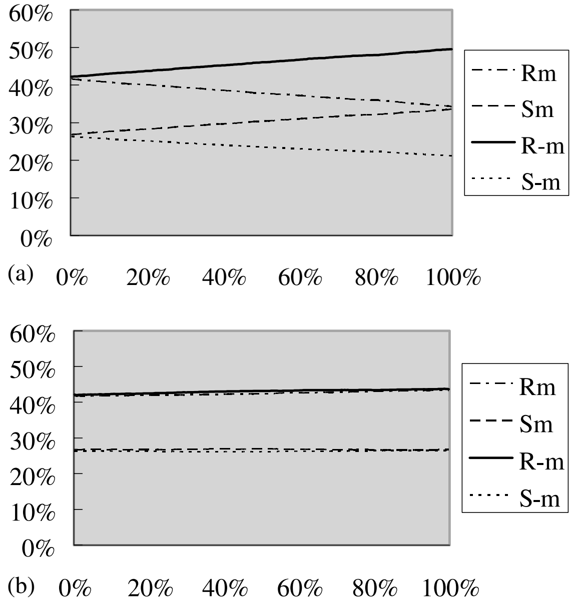

Fig. 8. RS-diagrams yielded by the dual statistics method by Fridrich et al. (2001) for stego-images produced by conventional LSB-embedding steganographic technique and our method.

5. Conclusions and suggestions

A new and efficient computer-based steganographic method for embedding secret messages into images without producing noticeable changes has been proposed. There is no need of referencing the original image when extracting the embedded data from a stego-image. The method utilizes the characteristic of the human vision's sensitivity to gray value variations. Secret data are embedded into a cover image 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 cover images with imperception, but also offers an easy way to accomplish secrecy. This embedding method can be easily extended to efficiently carry content-related messages such as captions or annotations in audios and videos by embedding data in each adjacent pair of signals of the data-streams.

Acknowledgements

This work is supported partially by National Science Council under Project NSC 89-2218-E-130-002 and partially by the MOE Program for Promoting Academic Excellence of Universities under the grant number 89-1-FA04-1-4.

Appendix A. Proof of Eq. (3)

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

$$(g'_i, g'_{i+1}) = \begin{cases} \left(g_i - \dfrac{m+1}{2}, g_{i+1} + \dfrac{m-1}{2}\right) & \text{when } g_{i+1} - g_i \text{ is odd and } m \text{ is odd;} \quad \text{(A.1)} \\ \left(g_i - \dfrac{m-1}{2}, g_{i+1} + \dfrac{m+1}{2}\right) & \text{when } g_{i+1} - g_i \text{ is even and } m \text{ is odd;} \quad \text{(A.2)} \\ \left(g_i - \dfrac{m}{2}, g_{i+1} + \dfrac{m}{2}\right) & \text{when } m \text{ is even,} \quad \text{(A.3)} \end{cases}$$

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}), i.e., 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 Eqs. (A.1)–(A.3) instead.

The proof of the correctness of Eq. (3) proceeds 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. (A.2) we have

$$f((g_i, g_{i+1}), m') = \left(g_i - \frac{m'-1}{2}, g_{i+1} - \frac{m'+1}{2}\right).$$

It is easy to see that the difference between $g_i - (m' - 1/2)$ and $g_{i+1} + (m' + 1/2)$ is odd.

Therefore,

$$\begin{aligned} f(f((g_i, g_{i+1}), m'), m'') &= f\left(\left(g_i - \frac{m'-1}{2}, g_{i+1} + \frac{m'+1}{2}\right), m''\right) \quad \text{by Eq. (A.2)} \\ &= \left(g_i - \frac{m'-1}{2} - \frac{m''+1}{2}, g_{i+1} + \frac{m'+1}{2} + \frac{m''-1}{2}\right) \quad \text{by Eq. (A.1)} \\ &= \left(g_i - \frac{(m'+m'')}{2}, g_{i+1} + \frac{(m'+m'')}{2}\right) \\ &= f((g_i, g_{i+1}), m' + m'') \quad \text{by Eq. (A.3)} \\ &= f((g_i, g_{i+1}), m) \end{aligned}$$

which is just the desired Eq. (3).

Secondly, if $g_{i+1} - g_i$ is odd, then by Eq. (A.1) we have

$$f((g_i, g_{i+1}), m') = \left(g_i - \frac{m'+1}{2}, g_{i+1} + \frac{m'-1}{2}\right).$$

It is easy to see that the difference between $g_i - (m' + 1)/2$ and $g_{i+1} + (m' - 1)/2$ is even.

Therefore,

$$
\begin{aligned}
f(f((g_i, g_{i+1}), m'), m'') &= f\left(\left(g_i - \frac{m'+1}{2}, g_{i+1} + \frac{m'-1}{2}\right), m''\right) \quad \text{by Eq. (A.1)} \\
&= \left(g_i - \frac{m'+1}{2} - \frac{m''-1}{2}, g_{i+1} + \frac{m'-1}{2} + \frac{m''+1}{2}\right) \quad \text{by Eq. (A.2)} \\
&= \left(g_i - \frac{(m'+m'')}{2}, g_{i+1} + \frac{(m'+m'')}{2}\right) \\
&= f((g_i, g_{i+1}), m'+m'') \quad \text{by Eq. (A.3)} \\
&= f((g_i, g_{i+1}), m)
\end{aligned}
$$

which is just the desired Eq. (3).

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

References

Anderson, R.J., Petitcolas, F.A.P., 1998. On the limits of steganography. IEEE J. Selected Areas Commun. 16, 474–481.

Artz, D., 2001. Digital steganography: Hiding data within data. IEEE Internet Comput. 5 (May/June), 75–80.

Bender, W., Gruhl, D., Morimoto, N., Lu, A., 1996. Techniques for data hiding. IBM Syst. J. 35 (3/4), 313–336.

Cox, I.J., Miller, M.L., Boom, J.A., 2000. Watermarking applications and their properties. In: Proc. Internat. Conf. on Information Technology: Coding and Computing, pp. 6–10.

Cox, I.J., Bloom, J., Miller, M., 2001. Digital Watermarking: Principles & Practice. Morgan Kaufmann Publishers, San Francisco.

Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T., 1997. Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. 6, 1673–1687.

Davern, P., Scott, M., 1996. Fractal based image steganography. In: Proc. First Internat. Workshop Information Hiding. In: Lecture Notes in Computer Science, Vol. 1174. Springer-Verlag, Berlin, pp. 279–294.

Franz, E., Pfitzmann, A., 2000. Steganography secure against cover-stego-attacks. In: Proc. Third Internat. Workshop Information Hiding. In: Lecture Notes in Computer Science, Vol. 1768. Springer-Verlag, Berlin, pp. 29–46.

Fridrich, J., Goljan, M., 2002. Practical steganalysis of digital images-state of the art. In: Proc. SPIE Photonics West, Vol. 4675, Conference on Security and Watermarking of Multimedia Contents, pp. 1–13.

Fridrich, J., Goljan, M., Du, R., 2001. Reliable detection of LSB steganography in grayscale and color images. In: Proc. ACM Workshop on Multimedia and Security, pp. 27–30.

Fridrich, J., 1998. Image watermarking for tamper detection. In: Proc. IEEE Internat. Conf. on Image Processing, Vol. II, pp. 404–408.

Fridrich, J., Rui, D., 2000. Secure steganographic methods for palette images. In: Proc. Third Internat. Workshop Information Hiding. In: Lecture Notes in Computer Science, Vol. 1768. Springer-Verlag, Berlin, pp. 61–76.

Hartung, F., Kutter, M., 1999. Multimedia watermarking techniques. Proc. IEEE 87, pp. 1079–1107.

Johnson, N.F., Jajodia, S., 1998. Exploring steganography: Seeing the unseen. IEEE Comput. (February), 26–64.

Koch, E., Zhao, J., 1995. Towards robust and hidden image copyright labeling. In: Proc. IEEE Nonlinear Signal and Image Processing Workshop, Thessaloniki, Greece, pp. 452–455.

Maxemchuk, N.F., 1994. Electronic document distribution. AT & T Tech. J. 73, 73–80.

Nikolaidis, N., Pitas, I., 1998. Robust image watermarking in the spatial domain. Signal Process. 66, 385–403.

Ohnishi, J., Matsui, K., 1996. Embedding a seal into a picture under orthogonal wavelet transform. In: Proc. Multimedia'96. IEEE Press, Piscataway, NJ, pp. 514–521.

Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G., 1999. Information hiding—a survey. Proc. IEEE 87, pp. 1062–1078.

Podilchuk, C.I., Delp, E.J., 2001. Digital watermarking: Algorithms and applications. IEEE Signal Process. Mag. 18 (4), 33–46.

Podilchuk, C.I., Zeng, W., 1998. Image-adaptive watermarking using visual models. IEEE J. Selected Areas Commun. 16, 525–539.

Ramkumar, M., Akansu, A.N., 2001. Capacity estimates for data hiding in compressed images. IEEE Trans. Image Process. 10, 1252–1263.

Swanson, M.D., Kobayashi, M., Tewfik, A.H., 1998. Multimedia data-embedding and watermarking technologies. Proc. IEEE 86, 1064–1087.

Turner, L.F., 1989. Digital data security system. Patent IPN, WO 89/08915.

Voyatzis, G., Pitas, I., 1999. The use of watermarks in the protection of digital multimedia products. Proc. IEEE 87, 1197–1207.

Walton, S., 1995. Image authentication for a slippery new age. Dr. Dobb's J.: Software Tools Profess. Program. 20, 18–26.

Westfeld, A., Pfitzmann, A., 2000. Attacks on steganography systems. In: Proc. Third Internat. Workshop Information Hiding. In: Lecture Notes in Computer Science, Vol. 1768. Springer-Verlag, Berlin, pp. 61–75.

Wolfgang, R.B., Podilchuk, C.I., Delp, E.J., 1999. Perceptual watermarks for digital images and video. Proc. IEEE 87, 1108–1126.

Yeung, M.M., Mintzer, F., 1997. An invisible watermarking technique for image verification. In: Proc. IEEE Internat. Conf. on Image Processing, Vol. II, pp. 680–683.

Zhao, J., Koch, E., Luo, C., 1998. Digital watermarking in business today and tomorrow. Commun. ACM 41, 67–74.