

Spatial-domain image hiding using image differencing

D.C.Wu and W.H.Tsai

Abstract: A method to embed a secret image into a cover image is proposed. The method is based on the similarity among the grey values of consecutive image pixels as well as the human visual system's variation insensitivity from smooth to contrastive. A stego-image is produced by replacing the grey values of a differencing result obtained from the cover image with those of a differencing result obtained from the secret image. The process preserves the secret image with no loss and produces the stego-image with low degradation. Moreover, a pseudorandom mechanism is used to achieve cryptography. It is found from experiment that the peak values of signal-to-noise ratios of the method are high and that the resulting stego-images are imperceptible, even when the size of the secret image is about a half of the cover image.

1 Introduction

Most text, image, audio, and video data can be represented in digital form. Owing to the human visual system's low sensitivity to small changes and the high plasticity of digital media, one can easily make small changes in digital data with low perceptibility, or even produce new data by mixing different data sources. Consequently, many interesting applications of data hiding in digital media can be created, like watermarking, caption data embedding, secret message delivery, etc. The purposes and requirements of the applications vary and are illustrated in Table 1.

Several previous works about data hiding in document images have been discussed in [1, 2], such as line-shift coding, word-shift coding, feature coding, etc. The quantity of the embedded data is small and can be easily destroyed by reproduction. In the fields of data hiding in images, the method of changing least significant bits (LSBs) [3, 4], the patchwork method [5], and the texture block coding method [2] all use the low sensitivity of the human visual system to small changes in grey values to embed data in the host image. Data hiding in images can also be conducted in the frequency or other transform domains. Such methods can embed a small amount of information in images with robustness: examples are the use of randomly sequenced pulse-position modulated codes [6], the secure spread spectrum method [7], wavelet-based embedding methods [8, 9], and DCT-based embedding methods [10, 11]. Incidentally, some terms about information hiding found in [12] are followed in this study.

We propose a new method to embed a grey-valued image into a grey-valued cover image. It can be used for delivering secret images such as confidential images, military maps, etc. The method is designed in such a way that the grey value of every pixel of the secret image is preserved, i.e. no distortion will be created in the secret image when it is extracted out from the stego-image. On the other hand, because the resulting stego-image usually contains a large quantity of embedded data, it may be degraded seriously. The proposed method, however, produces imperceptible changes in the resulting stego-image. Because the precision of pixel grey values may be destroyed when transforming them back and forth between the frequency and spatial domains, we adopt a way of embedding the secret image into the cover image directly in the spatial domain. Since a large quantity of data is directly embedded in the spatial domain, our method is not robust against most image transformations, such as lossy compression, cropping, etc. This is a common problem of the LSB-like methods.

The grey value replacement method proposed by Liaw and Chen [13] embeds a secret image into a cover image byte after byte by finding for each pixel p in the secret image a pixel p' in the cover image whose grey value g' is close to that (say, g) of p and then replacing g with g' . This method suffers from a problem: in certain images with different histogram shapes (Figs. 1*d* and 1*e*), there might exist many pixels in the secret image which do not have suitable pixels in the cover image with similar grey values to allow grey value replacement. To create suitable pixels for replacement, an adjustment method was performed firstly on the pixel values of the cover image. The effect of the result is similar to that of the method of histogram specification [14], which adjusts the shape of the histogram of the cover image to match that of the secret image. As shown in Figs. 1*c* and Figs. 1*f*, the resulting cover image may have serious distortion after the adjustment. In such a case, it is impossible to produce a stego-image without noticeable changes. The method proposed in this paper may be instead employed to embed an image into a cover image easily, and changes in the cover image after embedding are imperceptible to casual views. Moreover, during the work of data hiding we walk through the cover image in

© IEE, 2000

IEE Proceedings online no. 20000104

DOI: 10.1049/ip-vis:20000104

Paper received 27th July 1999

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China

D.C. Wu is also with the Department of Information Management, Ming Chuan University, Taipei, Taiwan 111, Republic of China

Table 1: Applications of data hiding in digital media

Application	Purpose	Requirements		Robustness	Need of original image to recover
		Amount of data to be embedded	Difficulty of detection		
Watermarking	copyright protection	small	no	yes	yes/no
Caption embedding	adding information into digital data	large	no	no	no
Secret message delivery	sending a message without attention	large	yes	no	yes/no

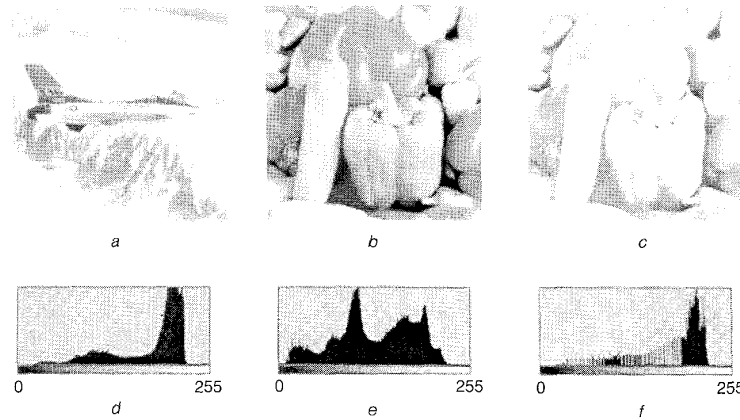


Fig. 1 Grey-value replacement of Liaw and Chen

- a 'Jet' image
- b 'Peppers' image
- c Result after performing histogram specification to 'Peppers' according to histogram of 'Jet'
- d Histogram of 'Jet'
- e Histogram of 'Peppers'
- f Histogram of resulting 'Peppers' after performing histogram specification

an order provided by a pseudorandom number generator to achieve cryptography, and so tampering access to the hidden data from illicit users can be prevented.

2 Proposed image hiding method

The method utilises the property of grey-value similarity among adjacent pixels which exists in most natural images (i.e. images not created by computers). The grey-value differences of adjacent pixels in quite different natural images may have probability density functions with very similar shapes. As an illustration, see the 256 grey-valued images of 'Jet' and 'Peppers' in Figs. 1a and 1b, respectively. The histograms of these two images are shown in Figs. 1d and 1e, respectively. They are quite different in shape. Now, perform the operation of 'image differencing' to the two images in the following way: for every pixel value g_i in either image I , where $i=1, 2, \dots$, let $g'_i = g_i - g_{i-1} + 128$ where g_{i-1} is the grey value of a neighbouring pixel to g_i , and create a new image I' with all g'_i as the new pixel values. Call I' the difference image from I . More details of image differencing in our proposed method, which are a little different from that just shown, are in the following section. The difference images from 'Jet' and 'Peppers' are shown in Figs. 2a and 2b, respectively, and their histograms are shown in Fig. 2c and 2d, respectively. The pixel values of the difference images are concentrated near 128 resulting from the grey-value simi-

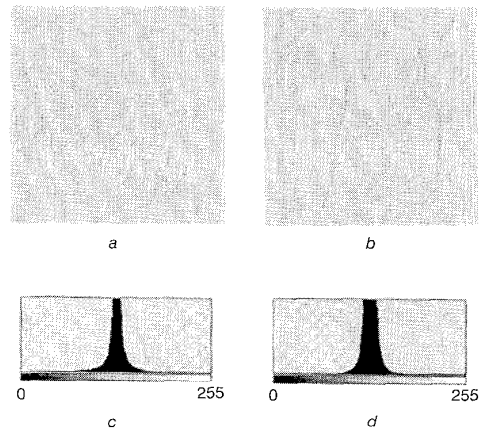


Fig. 2 Difference imaging

- a Difference image of 'Jet'
- b Difference image of 'Peppers'
- c Histogram of difference image of 'Jet'
- d Histogram of difference image of 'Peppers'

larity between adjacent pixels. The histograms of the difference images are quite similar in shape, in contrast with those of their original images, which are quite different. The similarity helps us exploit a way to embed a secret image easily by grey value replacement using a difference cover image and a difference secret image.

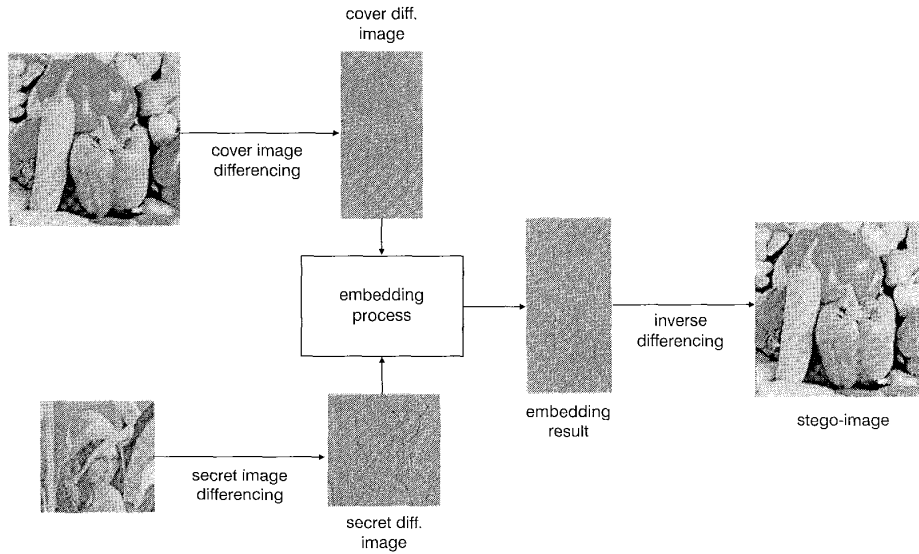


Fig. 3 Proposed image hiding method

Hiding data in images by replacing their LSBs is a simple embedding method that utilises the insensitivity of the human visual system to small changes in the image. The visual system's sensitivity indeed varies from smooth areas to contrastive areas. The amount of pixel-value modification which causes a just noticeable change in a smooth image area is smaller than that in an edge or contrastive area. Edge or contrastive areas can suffer greater changes with imperception. In the hiding process we propose, we use this characteristic to hide more data in contrastive areas and less in smooth areas. This helps to hide more data imperceptibly in an image.

The proposed hiding method is sketched in Fig. 3, which consists of three major parts: the process of cover image differencing, that of secret image differencing, and the embedding process, which are described subsequently.

2.1 Cover and secret image differencing

Cover and secret images used in the proposed method are assumed to have 256 grey values. To get a difference image from a given cover image, we obtain the grey value g of a pixel in the resulting image from every non-overlapping two-pixel subimage (p', p'') of the cover image, in a zigzag order as shown in Fig. 4, by

$$g = g'' - g' + 128 \quad (1)$$

where g' and g'' are the grey values of p' and p'' , respectively. The resulting data stream is just a half of the cover image in size.

The process of obtaining the difference image from the secret image is the same as that for the cover image except that we calculate the difference value of the grey values of every two adjacent pixels, instead of every non-overlapping two-pixel subimage, in zigzag order. So the size of the resulting stream is the same as that of the secret image. In the calculation of the difference value of the top-leftmost pixel of the secret image, we assume that its previous pixel's grey value is 128. For convenience, in the sequel we call the difference image from the secret image as the secret difference image, and that from the cover image as the cover difference image.

In addition, we quantise the resulting difference values into n ranges as illustrated in Fig. 5. These ranges are

indexed by 1 through n , respectively. A difference value which falls in a range with index k is said to have index k . The range intervals are selected in this study to meet the following criteria: all the values in a certain range (i.e. with an identical index) are considered as close enough, and if a value in the range is replaced by another in the same range, the change cannot be easily perceived. The range intervals of difference values we choose are based on the visual system's sensitivity variation from smooth to contrastive areas. The pixels in the contrastive area with difference values far from 128 may tolerate larger value changes than those in the smooth area with difference values near 128, when judged with the same sensitivity by human perception. So we create smaller ranges near 128 and larger ones far from 128 for the purpose of better replacement with

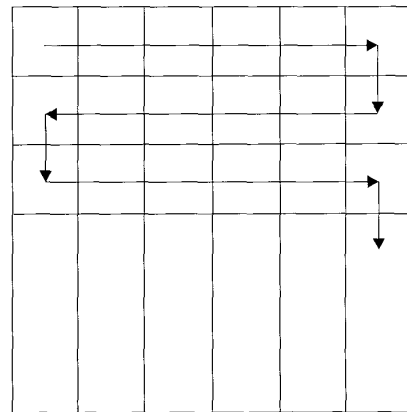


Fig. 4 Zig-zag order used for calculating difference values of both cover and secret images

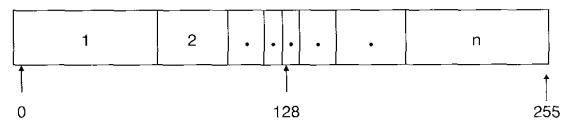


Fig. 5 Quantisation of difference values 0 to 255 into n unequal ranges

Index 1 to n is assigned to each of the ranges

less noticeable results. This strategy helps us to embed more information in contrast areas with less perception.

2.2 Image embedding process

The proposed image-embedding process consists of two steps: grey-value replacement, and the inverse differencing of the stego-image.

2.2.1 Grey-value replacement: An overview of the replacement process is given in Fig. 6. For each pixel p_s in the secret difference image S , we find a pixel p_c in the cover difference image C whose grey value has the same index as that of p_s , and hide p_s by replacing the value of p_c with that of p_s . Although the grey-value distributions of S and C are similar, there may exist insufficient pixels in C which have the same index as that of a certain pixel in S which we want to embed. So it is necessary to adjust the values of some pixels in C into the new values which are insufficient for embedding before the whole replacement work begins. The adjustment work starts by counting the total number of pixels of every grey value from 0 to 255 in images S and C , respectively, and computing the total count for each index. The total counts of indices from 1 to n for both S and C are then checked to find out 'insufficient indices'. We say that an index k is insufficient if the total count of the index k in C is smaller than that in S . The range of grey values that is represented by an insufficient index k is called an insufficient range. Ranges which are not insufficient are called excessive. For every insufficient range we need to accumulate enough pixels in C that are not with index k to complement the amount of insufficiency and adjust their indices to k by changing the values of the pixels to a value in the insufficient range. In this process, the adjustment work starts after the accumulation works of all insufficient ranges are processed. In the accumulation process we record only the number of pixels

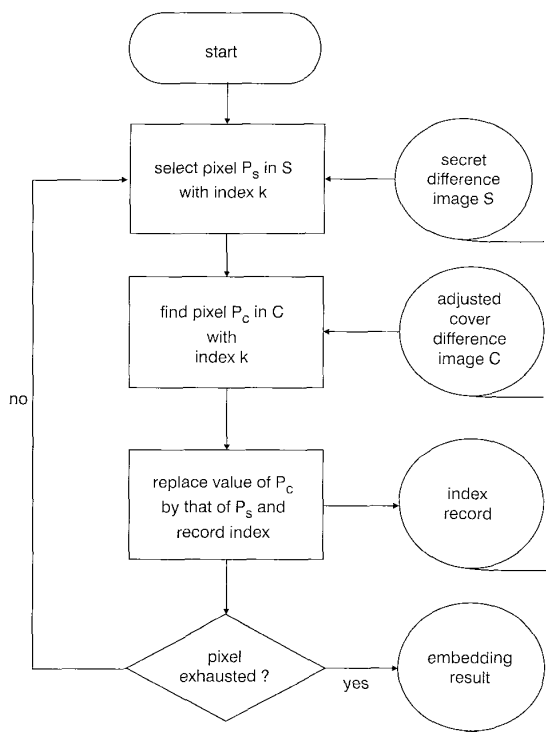


Fig. 6 Overview of grey value replacement process

of each grey value that is to be adjusted as well as the new grey value which the pixels will be adjusted to have. We accumulate for an insufficient range R a sufficient number of pixels by collecting the grey values spreading out of both ends of R . In this procedure we collect grey values from all of the ranges except the processed insufficient ranges. The number of pixels with a given grey value g which can be changed to have a new value in an insufficient range R is determined by the quantity of the need in R and the number of pixels with g . The accumulation work for R is finished when a sufficient number of pixels is collected. There may exist more than one insufficient range which need be processed. The order of processing the insufficient ranges in our method is decided by how large the distance of an insufficient range is to a nearest excessive range. To avoid the abrupt changes of grey values in some pixels, we apply the accumulation work to the insufficient range with the largest distance first.

After all the insufficient ranges are checked and the numbers of pixels to be changed are accumulated, the work of adjustment is performed by randomly traversing C using a pseudorandom generator, which visits each pixel in C only once. For every visited pixel in C with a grey value that needs to be changed, we change it to a value of the insufficient range according to a record obtained in the previous accumulation work. Since the adjustment work may cause more perceptible changes in some pixels, the random working mechanism provides a way to scatter the distortion over the whole image. After the adjustment process, the adjusted cover difference image is ready for use in subsequent replacement work.

For the purpose of easily finding the desired p_c in C which has the same index as p_s , we rearrange all the pixels in C into an n -list structure as illustrated in Fig. 7. Every list in the structure has an index which corresponds to the index of a grey-value range mentioned previously. Every node in the lists is a record of the location in C . A process of traversing every pixel of C is applied, in which we put the visited pixel into the rear of the corresponding list according to the index of its grey value. For example, if the location of 4866 in C with a value of 165 is visited and if the grey value belongs to the range whose index is k , then we add a node with the value 4866 to the rear of list k . Furthermore, we use a pseudorandom generator to permute the traversing order of the pixels in C instead of scanning through C sequentially. We use the permutation order to visit each pixel in C only once. This pseudorandom mechanism aims to achieve cryptography. This means that if an illicit user does not have the seed of the pseudorandom generator used in the image hiding process, the user cannot easily find out the correct traversing order. So,

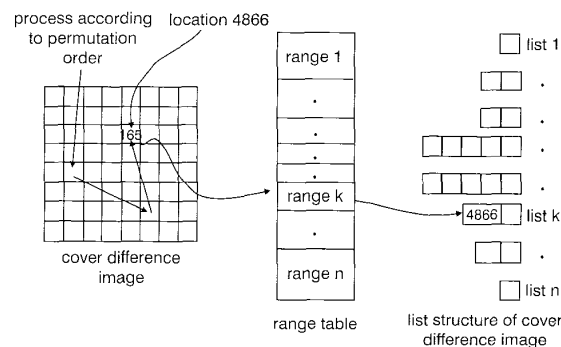


Fig. 7 Illustration of rearranging cover difference image data into list structure

the steps of extracting the hidden image cannot be followed successfully.

After constructing the list structure we find for every pixel in S the corresponding pixel in the list structure that has the same index to accomplish the replacement work. We process every pixel of S in the zigzag order mentioned previously. For each visited pixel p_s in S with index k , we extract the head element in the k th list of the list structure which denotes a location in C , and replace the grey value of this location with the grey value of p_s . For example (Fig. 8), if we visit a pixel in S with grey value 168 and index k , then we extract the head element from list k , whose value 4866 is a location in C where we want to do the replacement work. The grey value of the visited pixel is used to replace the grey value appearing in location 4866 of C , i.e. 165 is changed into 168. The replacement process is finished after all pixels in S are processed.

The grey value replacement process described is efficient because the one-to-one mapping between the pixels of the secret difference image and those of the cover difference image in the process can be built by traversing both the cover difference image and the secret difference image only once.

2.2.2 Inverse image differencing: In the inverse image differencing process which produces a stego-image, for each difference values d' in the processed cover difference image, an inverse calculation for this value is performed to find the grey values (g'_{i-1}, g'_i) of the corresponding two-pixel subimage (p'_{i-1}, p'_i) of the stego-image whose difference value is d' . Because we want to cause less perceptible distortion, the information about the grey values (g_{i-1}, g_i) of the corresponding two-pixel subimage (p_{i-1}, p_i) in the original cover image is needed. Assume the difference value of (g_{i-1}, g_i) is d . We produce (g'_{i-1}, g'_i) according to the following equations:

$$\begin{aligned} g'_{i-1} &= g_{i-1} - \left\lfloor \frac{d' - d}{2} \right\rfloor \\ g'_i &= g_i + \left\lfloor \frac{d' - d}{2} \right\rfloor. \end{aligned} \quad (2)$$

The two equations together satisfy the requirement that the difference of g'_{i-1} and g'_i is d' . Because the equations cause

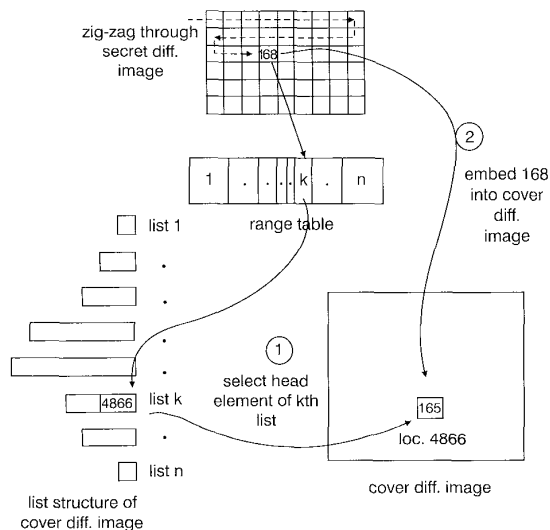


Fig. 8 Proposed process for embedding data into cover difference image

changes in g_{i-1} and g_i nearly equally to produce g'_{i-1} and g'_i , the distortion caused by changing g_{i-1} and g_i , is averaged over this two-pixel pair and so is less. Some of the calculations may cause g'_i or g'_{i-1} to fall off the boundary of the range $[0, 255]$ of a pixel value. In such cases we set the pixel value to the boundary value, i.e. to 0 or 255, and readjust the other pixel value to a new value to preserve the difference value of g'_{i-1} and g'_i to be d' .

2.2.3 Embedding of leading information: Since an extra table is constructed to record the index of the grey value of each pixel during the replacement process, the table must be available and used in the process of extracting the secret image from the stego-image. One can leave the table in a separate place or just hide it into the cover image. The latter way needs extra processing, in which we use the Huffman coding method to reduce the size of the table because the indices in our proposed method, with the values 1 to n , concentrate on just a few values. More specifically, we embed into the cover image the following extra information: (a) the width and height of the secret image, (b) the number of quantised ranges, the boundaries of each range, the number of elements in each quantised range, which are needed to construct the Huffman tree, and finally (c) the table of indices, which is organised as a bit stream generated by the Huffman coding method. This leading information provides a way to use different numbers of ranges and different range boundaries in the embedding process. The entire leading information, taken as a bit stream, is embedded into the LSBs of the cover image randomly. We walk through the cover difference image using a pseudorandom number generator and embed every six bits of the bit stream into a pixel-pair of the original cover image, i.e. each pixel in this pixel-pair takes three bits as the rightmost bits. After embedding the bit stream we use the rest of the cover difference image to hide the secret difference image, as described previously.

3 Process of extracting hidden image

The process of extracting the hidden image is proceeded by using first the seed of the pseudorandom number generation to extract out the data that are embedded in the stego-image. A stego-difference image S_d is produced from the stego-image S using a method similar to that for producing the cover difference image in the hiding process, and the seed of pseudorandom number generation is used to produce the same traversing order for visiting S_d as in the embedding process.

More specifically, we extract the leading information embedded in the stego-image directly from the three LSBs of the grey values of each pixel-pair in S that corresponds to the visited pixel of S_d . The grey values of the embedded secret difference image are extracted then by visiting the rest of the pixels of S_d . We rearrange S_d into an n -list structure as in the embedding process. The same list used in the embedding process is reconstructed by the same traversing order. And then the list of the range indices and the n -list structure are used in extracting the grey values from S_d to build the secret difference image. In short, the secret image can be recovered by applying an inverse differencing process to the secret difference image.

4 Experimental results

In our experiments four cover images are used, as shown in Fig. 9, each with size 512×512 , and the reduced-sized

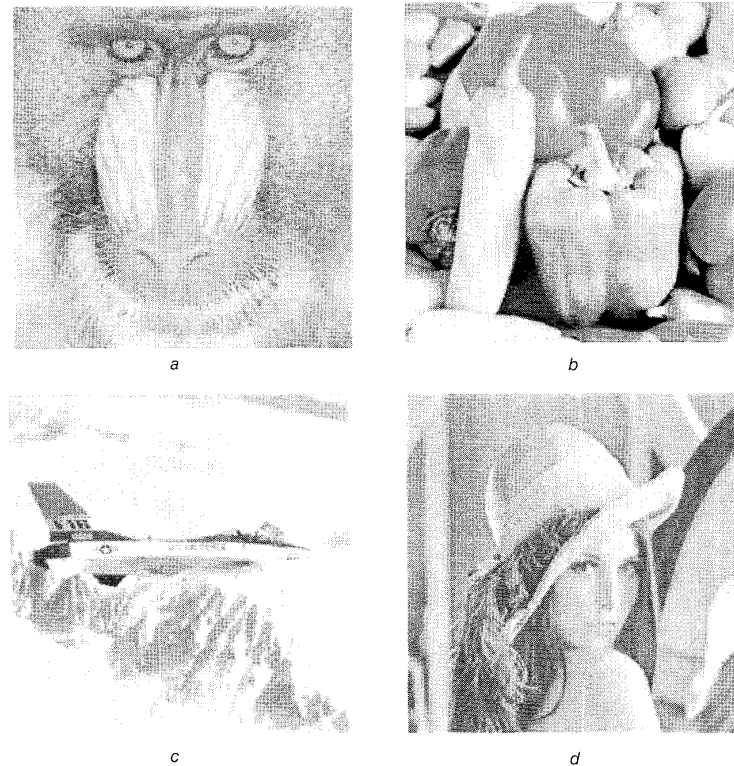


Fig. 9 Cover images (512 × 512)

- a 'Mandrill'
- b 'Peppers'
- c 'Jet'
- d 'Lena'

version of these images are used as the secret images, each with two sizes 256×512 and 256×256 . The mean μ and the standard derivation σ of the grey values of the pixels in each image are shown in Table 2. It is seen that the μ values of the images are quite different but those of the difference images are all close to 128 and the σ value of each difference image is smaller than that of the original image. Since the histograms of the difference images are similar in shape and the σ values are small, the effort to find similar pixel values in performing grey value replacement using the difference cover image and the difference

secret image is less than that using the original cover image and the original secret image.

A simple LSB-replacing experiment was conducted for the purpose of comparison. We embedded each secret image into the four cover images by directly replacing the four LSBs of the grey values of the pixels of the cover images, respectively. Also, four experiments were conducted to test the proposed method in which two sets of ranges were used for quantising the difference values g for the purpose of comparison. The ranges are illustrated in Table 3. The experiments so are divided into two groups by

Table 2: Means and standard derivations of grey values of cover images, cover difference images, secret images, and secret difference images

	Cover image (512 × 512)				Cover difference image (256 × 512)			
	Mandrill	Jet	Lena	Peppers	Mandrill	Jet	Lena	Peppers
Mean μ	135.45	179.28	130.32	127.15	128.03	128.00	127.99	128.00
Std ρ	38.99	46.68	44.10	52.84	20.13	12.43	10.46	11.62
	Secret image (256 × 512)				Secret difference image (256 × 512)			
	Mandrill	Jet	Lena	Peppers	Mandrill	Jet	Lena	Peppers
Mean μ	135.43	179.46	130.37	127.18	127.99	127.98	128.00	128.00
Std ρ	38.97	46.59	44.10	52.64	28.66	18.57	16.89	15.01
	Secret image (256 × 256)				Secret difference image (256 × 256)			
	Mandrill	Jet	Lena	Peppers	Mandrill	Jet	Lena	Peppers
Mean μ	135.48	179.34	130.36	127.43	128.00	128.02	127.98	128.00
Std ρ	35.57	46.21	44.11	52.76	18.77	16.91	16.89	15.85

Table 3: Two sets of quantisations of difference values and corresponding ranges and sizes used in experiments (top: range table 1; bottom: range table 2)

Index	1	2	3	4	5	6	7	8			
Range	0–71	72–103	104–119	120–127	128–135	136–151	152–183	184–255			
Size	72	32	16	8	8	16	32	72			
Index	1	2	3	4	5	6	7	8	9	10	11
Range	0–45	46–77	78–99	100–113	114–123	124–131	132–141	142–155	156–177	178–209	210–255
Size	46	32	22	14	10	8	10	14	22	32	46

using different range tables. We embedded each 256×512 secret image into the cover images without the leading information, and then embedded each 256×256 secret image into the cover images with the leading information.

The peaks of the signal-to-noise (PSNR) values of the five experiments are shown in Table 4. The resulting stego-images of the methods that use ‘Jet’ as the cover image and ‘Peppers’ as the secret image are selected and shown in Fig. 10, and those that use ‘Lena’ as the cover image and ‘Mandrill’ as the secret image are shown in Fig. 11. It is seen that changes of the LSBs may cause noticeable changes in smooth areas like the background of ‘Jet’ (Fig. 10a) and the shoulder of ‘Lena’ (Fig. 11a). The PSNR values of the results of our proposed method are almost all higher than that of the LSB-replacing method except in some of the results from using ‘Mandrill’ as the cover image or the secret image. This is because that the value σ of the difference image of ‘Mandrill’ is larger than

those of the other images, and so more quantity of grey value changes are needed during the embedding steps. The selected stego-images in Figs. 10 and 11 show that the results of the proposed method have less noticeable changes than those of the LSB replacing method, even when the PSNR is poor. Generally, the PSNR values coming from using the second range table are higher than those of using the first range table owing to the smaller range sizes of the second range table. The amounts of data resulting from embedding a 256×256 image plus its leading information is smaller than that resulting from embedding a 256×512 image without the leading information. The PSNR values of the latter are smaller than those of the former.

It is found that the embedding results of the proposed method are similar to the cover images. The quality is good even in a stego-image with a poor PSNR. This demonstrates that the changes made in edge areas are not noticeable.

Table 4: PSNR values of embedding secret images into cover images

Cover image	Embedding method			Secret image†			
				Mandrill	Peppers	Jet	Lena
Mandrill	Replacing four LSBs of pixels			32.68	32.41	31.96	32.58
	proposed method	without leading info.	range table 1	32.65	32.05	32.24	33.77
		with leading info.	range table 2	33.87	32.26	33.00	33.97
		without leading info.	range table 1	39.08	39.98	40.22	39.87
		with leading info.	range table 2	40.44	41.12	40.96	41.00
Peppers	Replacing four LSBs of pixels			32.72	32.46	31.96	32.61
	proposed method	without leading info.	range table 1	28.05	37.36	33.82	35.82
		with leading info.	range table 2	28.45	38.59	34.68	36.93
		without leading info.	range table 1	37.01	39.80	38.81	39.09
		with leading info.	range table 2	36.15	41.15	39.89	40.28
Jet	Replacing four LSBs of pixels			32.67	32.41*	31.99	32.57
	proposed method	without leading info.	range table 1	28.05	38.69*	35.25	36.41
		with leading info.	range table 2	28.45	40.53*	36.42	37.80
		without leading info.	range table 1	34.61	40.05*	40.19	39.76
		with leading info.	range table 2	34.76	41.58*	40.85	39.29
Lena	Replacing four LSBs of pixels			32.64*	32.38	31.87	32.52
	proposed method	without leading info.	range table 1	28.02*	37.47	33.96	35.75
		with leading info.	range table 2	28.31*	38.35	34.53	36.54
		without leading info.	range table 1	37.25*	39.44	38.99	39.17
		with leading info.	range table 2	37.03*	40.31	39.77	39.94

† Secret images are of size 256×512 except those used in proposed method of embedding with leading information (with size 256×256).

* Resulting stego-images are shown in Figs. 10 and 11.



Fig. 10 Resulting images of experiments of several methods that use 'Jet' as cover images and 'Peppers' as secret images

- a* Result by replacing four LSBs of pixels
- b* Result of proposed method without embedding leading information and using first range table
- c* Result of proposed method without embedding leading information and using second range table
- d* Result of proposed method embedding leading information and using first range table
- e* Result of proposed method embedding leading information and using second range table

5 Conclusion

We have proposed a novel method for embedding a grey-valued secret image into a cover image and preserving the secret image with no loss. The method produces the stego-image without noticeable changes. Image differencing operations are employed to create difference images from which similar values can be found easily in doing grey

value replacement. The method utilises the characteristic of the human visual system's sensitivity to embed more secret data. The method not only provides a way for embedding large quantities of data imperceptibly into cover images, but also offers an easy way to accomplish cryptography. In the future, we will study the problems of embedding images without extra tables and embedding images into colour images.

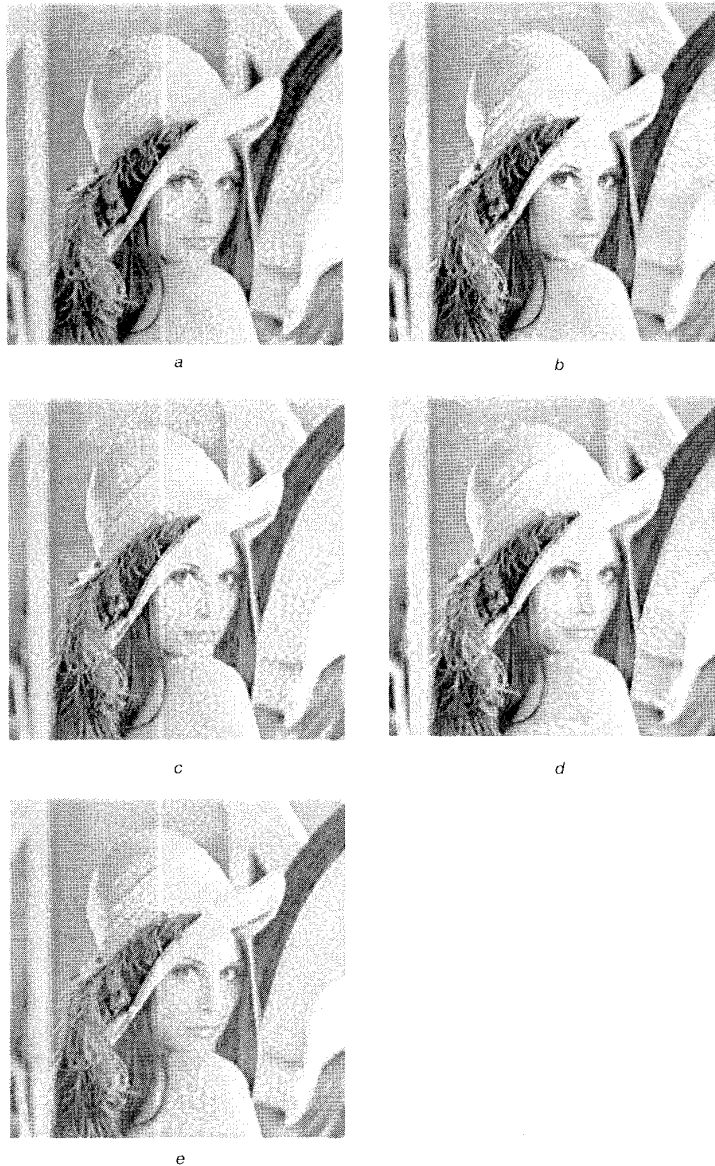


Fig. 11 Resulting images of experiments of several methods that use 'Lena' as cover images and 'Mandrill' as secret images

- a Result by replacing four LSBs of pixels
 b Result of proposed method without embedding leading information and using first range table
 c Result of proposed method without embedding leading information and using second range table
 d Result of proposed method embedding leading information and using first range table
 e Result of proposed method embedding leading information and using second range table

6 Acknowledgment

This work is supported partially by National Science Council, R. O. C. under grant NSC86-2213-E009-113.

7 References

- BRASSIL, J.T., LOW, S., MAXEMCHUK, N.F., and O'GORMAN, L.: 'Electronic marking and identification technique to discourage document copying', *IEEE J. Sel. Areas Commun.*, 1995, **13**, (8), pp. 1495–1503
- BENDER, W., GRUHL, D., MORIMOTO, N., and LU, A.: 'Techniques for data hiding', *IBM Syst. J.*, 1996, **35**, (3/4), pp. 313–336
- TURNER, L.F.: 'Digital data security system, 1989, Patent IPN WO 89/08915
- WALTON, S.: 'Image authentication for a slippery new age', *Dr. Dobbs J.*, 1995, **20**, (4), pp. 18–26
- BENDER, W., MORIMOTO, N., and GRUHL, D.: 'Method and apparatus for data hiding in images, 1997, US patent 5689587
- KOCH, E., and ZHAO, J.: 'Towards robust and hidden image copyright labeling'. Proceedings of IEEE workshop on *Nonlinear signal and image processing*, 1995, Thessaloniki, Greece, pp. 452–455
- COX, I.J., KILIAN, J., LEIGHTON, F.T., and SHAMOON, T.: 'Secure spread spectrum watermarking for multimedia', *IEEE Trans. Image Process.*, 1997, **6**, (12), pp. 1673–1687
- OHNISHI, J., and MATSUI, K.: 'Embedding a seal into a picture under orthogonal wavelet transform'. Proceedings of Multimedia'96. (IEEE Press, Piscataway, NJ, 1995), pp. 514–521
- CHAE, J.J., and MANJUNATH, B.S.: 'A robust embedded data from wavelet coefficients', *Proc. SPIE-Int. Soc. Opt. Eng.*, 1998, **3312**, pp. 308–317
- HSU, C.T., and WU, J.L.: 'DCT-Based watermarking for video', *IEEE Trans. Consum. Electron.*, 1998, **44**, pp. 206–216
- BARNI, M., BARTOLINI, E., CAPPPELLINI, V., and PIVA, A.: 'A DCT-domain system for robust image watermarking', *Signal Process.*, 1998, **66**, pp. 357–372
- PFITZMANN, B.: 'Information hiding terminology. Proceedings of first int'l workshop on 'Information hiding'. Lecture Notes in Computer Science' no. 1174 (Springer-Verlag, Berlin, 1996), pp. 347–349
- LIAW, M.S., and CHEN, L.H.: 'An effective data hiding method'. Proceedings of IPPR conference on 'Computer vision, graphics, and image processing', 1997, Taiwan, R.O.C., pp. 146–153
- GONZALEZ, R.C., and WOODS, R.E.: 'Digital image processing' (Addison Wesley, New York, 1992)



• Online Journals •

The IEE Online Journal service is available direct from the IEE over the WorldWide Web.

<http://ioj.iee.org.uk/>

This service includes the IEE's rapid-publication letters journal, *Electronics Letters*, and all 12 titles of *IEE Proceedings*. Articles are stored in PDF format and may be viewed and printed using the Adobe® Acrobat® Reader.

Features

- Cross-journal searching of bibliographic records (inc. authors, titles, abstracts)
- Browsable table of contents pages for each journal issue
- Typeset quality display of articles from mid-1996
- Backfile from 1994 to mid-1996 based on scanned images
- Ability to print articles locally
- Updates in advance of print publication

Access Method

- Internet—networked machines or dial access via SLIP/PPP
- Netscape Navigator 2.0 (or higher) or other compatible browser
- Adobe Acrobat Reader

Pricing

The IEE Online Journal subscriptions are the same price as the equivalent full print subscriptions. This covers unlimited online searching, display and local printing by users at a single site.

Print Subscriber Discounts

Discounts are available for combination print/online subscriptions and for multiple journal subscriptions.

New for the Millenium

IEE Conference Proceedings, Colloquium/Seminar Digests, IEE Review and IEE Engineering Journals will also be available online via the IOJ service.



For full information, send for the leaflet: 'Online Journals from the IEE'. Available from: The Marketing Department, IEE, Michael Faraday House, Six Hills Way, Stevenage, Herts. SG1 2AY, UK.
Tel: +44 (0)1438 313311 Fax: +44 (0)1438 742840 E-mail: inspec@iee.org.uk