이진 이미지에 대한 픽셀값 가중치를 이용한 자료 은닉 기법 연구

A Data Hiding Method of Binary Images Using Pixel-value Weighting

정 기 현*
Jung, Ki-Hyun

ABSTRACT

This paper proposes a new data hiding method for binary images using the weighting value of pixel-value differencing. The binary cover image is partitioned into non-overlapping sub-blocks and find the most suitable position to embed a secret bit for each sub-block. The proposed method calculates the weighted value for a sub-block to pivot a pixel to be changed. This improves the image quality of the stego-image. The experimental results show that the proposed method achieves a good visual quality and high capacity.

주요기술용어(주제어) : Data Hiding(자료은닉), Information Hiding(정보은닉), Steganography(스테가노그래피), Binary Image(이진이미지)

1. Introduction

Data hiding is related to embed secret information in a cover data, such as image, audio and video and so on. This technique has much interest by growing the internet and developing of compression techniques.

The most important requirement in data hiding is that the detector could not detect the presence of the hidden data for the cover data.

Many techniques have been proposed for digital color and gray-scale images, most of them categorized as least significant bit(LSB) substitution method and pixel-value differencing(PVD) method. LSB substitution method is one common and well-known method of data hiding for gray or color images. For this method, the secret data is converted to match the fixed length of pixels. Other methods, based on using variable amounts of bits instead of fixed-lengths, have also been proposed. The pixel-value differencing method provides both high embedding capacity and good concealment. However, these methods cannot be directly applied to binary images. The difficulty lies in the fact that changing pixel values in a binary image can cause irregularities that are

† 2008년 5월 6일 접수~2008년 6월 20일 게재승인
* 영진전문대학 컴퓨터정보계열
주저자 이메일 : kingjung@paran.com

visually very noticeable. Hiding data in binary images is therefore more challenging than hiding secret data in other formats[1]. Not all of them can be directly applied to binary images in general.

On the other hand, very little research has been done on data hiding in binary images although the binary image is common and often appears in cartoons, newspapers, faxes, and magazines. Binary images are two-color images, with a 0 or 1 value for each pixel, in which each pixel requires only one bit to represent blank and white. It makes a difficulty to hide secret data in a binary image, which cause visually very noticeable. The measurement of stego-image is a large amount of data and visual quality as well as transparency and robustness. The peak signal -to-noise ration(PSNR) is used to judge the quality of the embedded image.

This paper proposes a new data hiding method for binary images using pixel-value weighting. The binary cover image is partitioned into non-overlapping sub-blocks and find the most suitable position to embed a secret bit for each sub-block. The proposed method calculates a weighted value for a sub-block to pivot a pixel to be changed.

This paper is organized as follows. Section 2 reviews the previous method which applied to binary images and Section 3 describes our proposed method. In Section 4, the experimental results are presented and discussed. Finally, the conclusions are presented in Section 5.

2. Related Work

In this section, the previous data hiding method, which embeds secret data to the binary image, is reviewed and introduces traditional technique of block masking in image processing.

2.1 Data Hiding for Binary Images

There are two primary methods of data hiding for binary images : sub-block modification and single pixel manipulation. The first modifies the sub-block, which is divided into a group of pixels. Matsui and Tanaka manipulated the dithering patterns to embed secret data and also embedded data in fax images by calculating the run lengths [2]. Low et al. changed line spacing and character spacing in textual images for bulk electronic publications[3,4] The second approach modifies a single pixel, from black to white or vice versa. Koch and Zhao proposed a data hiding method by forcing the ratio of black and white pixels in a block to be larger or smaller than one[5]. However, only a limited number of bits can be embedded, since the enforcing method has trouble dealing with blocks that contain a lower or higher black pixels. Wu et al. selected by calculating a characteristic value and finding a pattern to embed secret data[6]. Liu et al. partitioned the binary image into blocks of 2×2 pixels and embedded a secret bit in the block by modifying 0.5 pixels on average[7]. Wu and Liu manipulated the flappable pixels to embed secret data into shuffled blocks. The shuffling of the blocks before embedding ensures the equalization of the embedding capacity without noticeable visual effects[8]. Venkatesan et al. proposed using the parity of blocks. The cover image is partitioned into small blocks, in which one bit information is stored if all of the pixel values does not contain 0 or 1[9]. Pan et al. proposed a novel data hiding method by partitioning into 4×4 blocks, where each block was repartitioned into overlapping sub -blocks. Secret data are hidden in the center region of each block[10].

2.2 Detection of Edge for Binary Image

An edge implies intensity changes in grey-level images and intensity or chromaticity changes, or both, in color images. Edges are usually taken as an important feature of images and are widely applied to high-level applications, such as image compression, region segmentation, pattern recognition, image retrieval, data hiding, etc. Research on edge detection methods can be divided into two categories: the first detects edge boundaries in gray-level images, and the second is designed for edge detection in color images. Detecting edge boundaries in gray-level images is based on changes in the pixel intensities; and several such schemes have been proposed, including the Sobel, Roberts, Laplacian and Canny operators, which employ convolution masks to determine possible edge locations[11~15]. A color image, however, consists of RGB channels; thus a color edge detection process must consider changes in intensity, chromaticity, or both. Several color edge detection schemes already have been developed and can be classified into two different approaches. The first approach processes the three-channel image as three gray-level images, then uses a gray-level edge detection scheme to detect an edge image separately for each color channel, resulting in three edge images for the RGB channels. Finally, the approach executes a merging procedure to combine the edge images into a targeted edge image[16~18]. The second approach has a two-stage structure, in which the first stage applies a channel reduction technique to reduce the color image's dimensionality from three to one. The second stage applies an edge detection procedure to the reduced one-channel image to produce an edge image[19~21].

3. Proposed Method

In this Section, the embedding and extracting algorithm are described. For a cover image, partition into $n \times n$ sub-blocks and find the optimal position to embed a secret bit.

3.1 Data Embedding

The detailed algorithm for embedding secret data is described as follows.

Algorithm 1. The embedding process

Input : A cover-image of cw×ch and secret data and the pixel-value weighting keys.

Output : A stego-image of $sw \times sh$.

Step 1 : Divide into $n \times n$ non-overlapping sub-blocks.

Step 2 : Calculate a weighted value for each sub-block. Assume that n=3, weighted value for one pixel, $w=|x+y|$ and each pixel value is given as Fig. 1. In here, x and y are weighted values for two columns and two rows, which are given as (-1, -2, -1), (1, 2, 1), (1, 2, 1), and (-1, -2, -1) for each two-pair. For example, let $w1$, $w2$, ..., $w9$ for nine pixels. For pixel $w2$, (0, 0, 1), (1, 1, 0) pixels are belonged for two columns and (0, 1, 1), (1, 0, 0) pixels are for two rows, so $x=(-1)+1+2=2$ and $y=2+1+(-1)=2$. Finally, $w2=2+2=4$.

Step 3 : For a sub-block, calculate the sum of the pixel value, say S.

Step 4 : Embed a secret bit into a sub-block when S is not equal to zero or $n \times n$ value.

Case 1. If S mod 2=0 and a secret bit is 1, then operate exclusive-OR to the pixel value belonging to the position which contains the largest weighted value for a sub-block.

Case 2. If S mod 2=1 and a secret bit is 0, then operate exclusive-OR to the pixel value belonging to the position which contains the largest weighted value for a sub-block.

Case 3. Otherwise, any other value of pixel for a sub-block remains unchanged.

For example, assume that a secret bit is 0 for Fig. 1, since S=5, $w2$, which contains the largest weighted value, is changed to 0.

Step 5 : For the embedded sub-block, if the sum of the pixel value is equal to zero or nn value, discard this block for embedding a secret bit.

A sub-block of cover

1	0	1	1	1
0	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	0	0	0	1

Weighted Value

	0	4	2	
	4	0	2	
	2	0	2	

[Fig. 1] Weighted values

3.3 Data Extracting

The following steps are executed to recover the secret data. It can be directly extracted from the stego-image only.

Algorithm 2. The extracting process

Input : A stego-image of $cw \times ch$ and sub-block size information.

Output : The secret data.

Step 1 : Divide into $n \times n$ non-overlapping sub-blocks.

Step 2 : Calculate the sum of the sub-block.

Step 3 : For all sub-block, extract a secret bit to the following two cases.

Case 1. If S mod 2=0, extract 0 from the selected sub-block.

Case 2. If S mod 2=1, extract 1 from the selected sub-block.

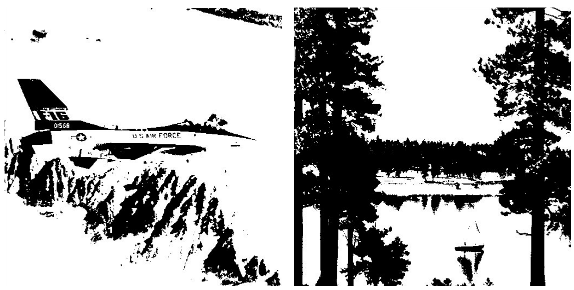
For above example, if S mod 2=0, then it can be extracted directly that the secret bit is 0.

Step 4 : Concatenate the extracted bits in order.

4. Experimental Results

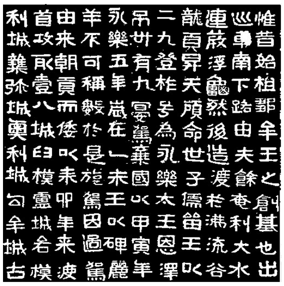
In our experiments, six 512×512 gray images shown in Fig. 2 were used as cover images, which a sub-block is divided into 3×3, and the secret data is generated by pseudo-random numbers. This paper adopts the peak signal-to-noise ratio(PSNR) to measure the imperceptibility and capacity for the amount of embedded data. To calculate PSNR, one bit of blank and white is extended to 8-bit blank and white value.

Table 1 shows that the results of the propose method for embeddable pixel position, embedded capacity and visual quality. If a sub-block is all black or white, it cannot be used to embed a secret bit and all of bits of a sub-block changed to black or white after embedding, it cannot also.



(a) Airplane (b) Boat


(c) Gatbawi (d) Peppers

(e) Lena (f) Epitaph

[Fig. 2] Cover images

[Table 1] Results of the proposed

Cover Images	The proposed method		
	Embeddable pixels	Capacity (bits)	PSNR (dB)
Airplane	3,879	2,876	22.85
Boat	4,737	3,283	21.42
Gatbawi	10,553	9,369	17.89
Peppers	3,685	2,866	22.83
Lena	4,656	3,403	21.61
Epitaph	10,033	7,606	18.45

Fig. 3 shows the embeddable blocks, which contains one or more pixel to be embedded. In our method, just only one bit can be embedded per a sub-block.


(a) Airplane (b) Boat


(c) Gatbawi (d) Peppers

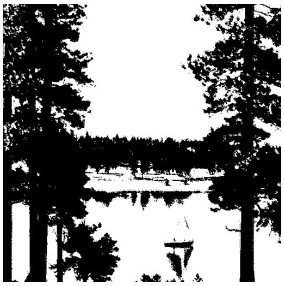

(e) Lena (f) Epitaph

[Fig. 3] Embeddable blocks

Fig. 4 shows the stego-images after embedding secret data. Since the embedded bit is mainly located in the boundary of blank and white, it is difficult to detect by human visual systems. Fig. 5 shows the effect on the cover image. For embeddable sub-blocks, one bit of secret data is hided per each sub-block.

Fig. 6 shows the embedded pixels, which were changed their bit value when embedding secret data.

(a) Airplane

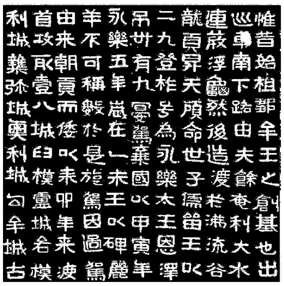


(b) Boat

(c) Gatbawi

(d) Peppers

(e) Lena

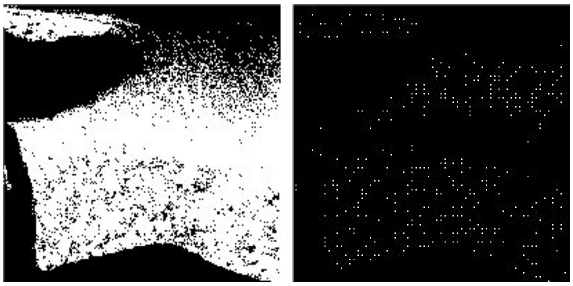


(f) Epitaph

[Fig. 4] Stego-images

(a) Partial cover image

(b) Partial embeddable blocks



(c) Stego-image and embedded pixels

[Fig. 5] Embedding effect

(a) Airplane

(b) Boat

(c) Gatbawi

(d) Peppers

(e) Lena (f) Epitaph

[Fig. 6] Embedded pixels

5. Conclusions

We have proposed a data hiding method for binary images using the weighting value of pixel-value differencing. The proposed method divided a cover image into sub-block and found the most suitable position to embed a secret bit. For a 3×3 sub-block, weighted values of each pixel were calculated given 3×3 masking values already, and then the largest weighted value is pivoted as embedding one bit of secret data. The experimental results have shown that the proposed method could provide a high capacity and maintain a good visual quality.

References

[1] H. Liang, W. Ran, X. Nie, “A Secure and High Capacity Scheme for Binary Images”, Proc. of the ICWAPR, 224～229, 2007.

[2] Z. W. C. Liu, Y. Dai, “A Novel Information Hiding Method in Binary Images”, Journal of Southeast University, 2003.

[3] J. Z. E. Koch, “Embedding Robust Labels into Images for Copyright Protection”, Proc. of the International Congress on Intellectual Property Rights for Specialized Information, 1995.

[4] Y. J. W. G. Pan and Z. H. Wu, “A Novel Data Hiding Method for Two-Color Images”, LNCS, pp. 261～270, 2001.

[5] N. F. M. J. T. Brassil, S. H. Low, “Copyright Protection for the Electronic Distribution of Text Documents”, Proc. of IEEE, pp. 1181～1196, 1999.

[6] K. T. K. Matsui, “Video-Steganography : How to Secretly Embed a Signature in a Picture”, Proc. IMA Intellectual Property Project, pp. 187～206, 1994.

[7] K. D. K. T. M. Venkatesan, P. Meenakshidevi, “A New Data Hiding Scheme with Quality Control for Binary Images using Block Parity”, 3rd Inter. Symposium on Information Assurance and Security, pp. 468～471, 2007.

[8] B. L. M. Wu, “Data Hiding in Binary Image for Authentication and Annotation”, IEEE Trans. on Multimedia, pp. 528～538, 2004.

[9] B. L. M. Wu, E. Tang, “Data Hiding in Digital Binary Images”, IEEE Inter. Conf. on Multimedia and Expo, 2000.

[10] A. M. L. S. H. Low, N. F. Maxemchuk, “Document Identification for Copyright Protection using Centroid Detection”, IEEE Trans. on Comm., pp. 372～383, 1998.

[11] P. Tsai, C. C Chang and Y. C. Hu, “An Adaptive Two-Stage Edge Detection Scheme for Digital Color Images”, Real Time Imaging 8(4), pp. 329～343, 1985.

[12] W. K. Pratt, “Digital Image Processing Wiley”, 1991.

[13] J. F. Canny, “A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 679～698, 1986.

[14] L. S. Davis, “A Survey of Edge Detection Techniques”, Computer Vision Graphics and Image Processing, pp. 248～270, 1975.

[15] D. Geman, “Stochastic Model for Boundary

Detection", Image Vision Computing, pp. 61 ~65, 1987.

[16] S. D. Zenzo, "A Note on the Gradient of a Multi-Image", Computer Vision, Graphics and Image Processing, pp. 116~128, 1986.

[17] R. Nevatia, "A Color Edge Detection and its Use in Scene Segmentation", IEEE Transactions on System and Man and Cybernetic, pp. 802~826, 1997.

[18] C. J. Delcroix and M. A. Abidi, "Fusion of Edge Maps in Color Images", Proceedings of SPIE International Society Optical Engineering, Vol. 1001, pp. 545~554, 1988.

[19] C. C. Chang, T. S. Chen and Y. Lin, "An Efficient Edge Detection Scheme of Color Image", Proceedings of the Fifth Joint Conference on Information Science, pp. 448~455, 2000.

[20] C. K. Yang and W. H. Tsai, "Reduction of Color Space Dimensionality by Moment-Preserving Thresholding and its Application for Edge Detection in Color Image", Pattern Recognition Letters, pp. 481~490. 1996.

[21] S. C. Pei and C. M. Cheng, "Color Image Processing by using Binary Quaternion Moment Preserving Thresholding Technique", IEEE Transactions on Image Processing, pp. 614~628, 1999.