Hiding an Image inside another Image using Variable-Rate Steganography

Abdelfatah A. Tamimi
Dept. of Computer Science
Faculty of Science & I.T.
Al-Zaytoonah University of Jordan
Amman, Jordan

Ayman M. Abdalla
Dept. of Multimedia Systems
Faculty of Science & I.T.
Al-Zaytoonah University of Jordan
Amman, Jordan

Omaima Al-Allaf
Dept. of C.I.S.
Faculty of Science & I.T.
Al-Zaytoonah University of Jordan
Amman, Jordan

***Abstract*—A new algorithm is presented for hiding a secret image in the least significant bits of a cover image. The images used may be color or grayscale images. The number of bits used for hiding changes according to pixel neighborhood information of the cover image. The exclusive-or (XOR) of a pixel's neighbors is used to determine the smoothness of the neighborhood. A higher XOR value indicates less smoothness and leads to using more bits for hiding without causing noticeable degradation to the cover image. Experimental results are presented to show that the algorithm generally hides images without significant changes to the cover image, where the results are sensitive to the smoothness of the cover image.**

Keywords—image steganography; information hiding; LSB method

I. INTRODUCTION

Steganography is a method of hiding a secret message inside other information so that the existence of the hidden message is concealed. Cryptography, in contrast, is a method of scrambling hidden information so that unauthorized persons will not be able to recover it. The main advantage steganography has over cryptography is that it hides the actual existence of secret information, making it an unlikely target of spying attacks. To achieve higher security, a combination of steganography with cryptography may be used.

In this paper, a new algorithm is presented to hide information in the least significant bits (LSBs) of image pixels. The algorithm uses a variable number of hiding bits for each pixel, where the number of bits is chosen based on the amount of visible degradation they may cause to the pixel compared to its neighbors. The amount of visible degradation is expected to be higher for smooth areas, so the number of hiding bits is chosen to be proportional to the exclusive-or (XOR) of the pixel's neighbors. Analysis showed effectiveness of the algorithm in minimizing degradation while it was sensitive to the smoothness of cover images.

II. BACKGROUND AND RELATED WORK

Surveys of different steganography techniques were presented in previous work, where secret information may be hidden in text, audio, image or video [1], [2], [3], [4], [5], [6], [7]. When an image is chosen to be used for hiding information, it is called a cover image. A cover image containing the secret information is called a stego image.

Hiding in LSBs of each pixel is desired since their modification will cause less distortion compared to other bits. The number of bits used should be variable and related to the stego image to minimize distortion [8], [9]. However, some applications, such as lossy compression, involve image alteration where some LSBs are lost. In such cases, more significant bits are used by transformation algorithms that utilize the special features of these applications. These techniques generally append coding information to the image with minimal or no change to the original pixels [10], [11].

Generally, the related previous work did not focus on hiding images inside other images. In addition, related image steganography research was usually limited to either grayscale or Red-Green-Blue images; not generalized to work for both image types. The new algorithm of this paper handles hiding different images inside other images of various types.

III. THE HIDING ALGORITHM

This algorithm uses a variable number of LSBs from each pixel of the cover image for hiding. A grayscale image consists of only one color matrix. A Red-Green-Blue (RGB) color image consists of three matrices representing the three colors. The number of bits chosen from each pixel color (red, green, and blue) is different. Images in other color formats may be converted to RGB matrices and converted back after the hiding process is done. The actual number of bits changes according to neighborhood information of each pixel color. When the resemblance between the neighbors of a pixel color entry is low, the pixel entry is located in a non-smooth area where change will not be detected easily. Therefore, the number of bits used for hiding is chosen to be proportional to the neighbors' XOR value for each pixel color entry.

The pixels used in hiding are those located in every line and every other column of the cover image, as in the white squares of a chess board. Pixels on the borders are not used for hiding. This means that approximately 50% of the pixels are used for hiding, while the rest of the pixels are used in determining hiding values and hiding capacity. For RGB images, each color is treated separately. The hiding process starts with the Red matrix, followed by the Green, and then the Blue. The XOR is computed for the value of each one of these pixels' four neighbors: left, right, above, and below. This comparison measures the smoothness of the pixel's neighborhood so that the number of hiding bits can be determined.

The algorithm for hiding in each color matrix is shown in Fig. 1, where *stegoC* is *stegoR*, *stegoG*, or *stegoB*, corresponding to the Red, Green, and Blue matrices of the original stego image, respectively. Each of these matrices has the same ($n \times m$) dimensions as the original image. In grayscale images, *stegoC* is the single color matrix. This algorithm takes each color matrix individually, and it goes through every line of the matrix starting with the second line and stopping at the line before the last. It goes through the entries in every other column, taking odd and even numbered columns in odd and even numbered lines, respectively. Left and right border columns are not used for hiding. The XOR of the four neighbors of each examined entry is computed. If the XOR value is less than a given threshold (α), only one LSB is used for hiding. Otherwise, the number of LSBs (*numLSBs*) used will be the ceiling of one-half of the XOR value. In the implementation of this paper, α was set to 9 and the maximum number of LSBs used for hiding in any pixel color was 4. To enhance avoidance of detection for RGB hidden images, avoid grouping all color information of a hidden pixel in a single location in the stego image.

The extraction process searches each of the three color matrices (Red, Green, and Blue), going through all lines and every other column as in the hiding procedure. The number of bits used for hiding in an entry, *stegoC*(*row*, *col*), is also determined by examining *x*; the XOR of the four neighbors as in the hiding process. All extracted hidden values are concatenated and grouped into bytes to form the original secret image.

```
row = 2
while (row ≤ n-2) and (the secret image is not finished)
  col = 2 + (row MOD 2)
  while col ≤ m-2
    x = stegoC(row-1,col) ⊕ stegoC(row+1,col) ⊕ stegoC(row,col-1)
      ⊕ stegoC(row,col+1))
    if x ≤ α
      numLSBs = 1
    else
      numLSBs = ⌈x/2⌉
    endif
    replace LSBs of stegoC(row,col) with the next numLSBs bits
      from the secret image
    col = col + 2
  endwhile
  row = row + 1
endwhile
```

Fig. 1. Algorithm for hiding in one color matrix.

IV. Implementation and Analysis

The algorithm was applied using 35 different images of different types and sizes for hiding. The sizes of these secret images ranged from 55×110 to 175×148 pixels. Three different cover images were used: Valley (2560×1920 pixels), Street

(a) Face image

(b) Original Valley image

(c) Original Street image

(d) Original Office image

Fig. 2. Original hidden and cover images.

(1920×2560 pixels), and Office (3001×2375 pixels). These cover images were chosen for having different smoothness characteristics where the Office image has visibly more smooth areas than the other two images.

The analysis of the results focus on two aspects: difficulty to detect the hidden image existence in the stego image and sensitivity to the smoothness of the cover image. Recall that only non-adjacent pixels are used for hiding. These are approximately 50% of the pixels in the image.

Fig. 2 shows one sample secret image (Face), which is 148×175 pixels, and the three cover images. Fig. 3 shows the three stego images where each of them is hiding a copy of the Face image. As seen in the figures, the difference between the original images and the stego images is not visible to the human eye. TABLE I shows the measurements obtained for these three stego images, where the percentage values show the ratios for using 1, 2, 3, or 4 bits per pixel color entry for hiding. Recall that RGB images have three color entries per pixel, compared to one entry in grayscale images. The peak signal-to-noise ratio (PSNR) and correlation values were the highest for the Office cover image. This cover image has mostly smooth areas, which caused the algorithm to choose only one bit for hiding in each of 84.6% of the pixel entries used for hiding, as seen in TABLE I. The other two cover images used more bits per entry, where Valley used more entry bits than Street.

(a) Stego Valley image

(b) Stego Street image

(c) Stego Office image

Fig. 3. Stego images after hiding the Face image.

The average results for all 35 test images are shown in TABLE II. The average correlation value was taken for the absolute values of correlation for all images, where the original cover image was compared to each of its stego images to obtain the individual correlation values.

As TABLE II shows, the PSNR and correlation values were high, indicating low degradation of stego images and big difficulty for hidden image detection. The correlation and PSNR values were the highest for the Office cover image. This mostly-smooth cover image caused the algorithm, on average, to choose only one bit for hiding in each of 85.8% of the pixels used for hiding, as seen in TABLE II. The other two cover images used more bits per entry, where Valley used more bits than Street. This indicates that images with smoother areas are a poor choice for cover images since they must use fewer bits for hiding to avoid detection, consequently lowering their hiding capacity. The slight increase in PSNR and correlation values for such images may not be a feasible expense for the significant decrease of hiding capacity.

V. CONCLUSIONS

The new algorithm presented in this paper uses a variable number of LSBs from each color of each considered pixel for hiding a secret image, where approximately 50% of all pixels are considered for hiding. The actual number of hiding bits in a pixel is inversely proportional to the smoothness of its neighbors. The smoothness of a pixel area is determined by taking the XOR of the pixel's neighbors, where a high XOR value indicates less smoothness.

Test results showed that the new algorithm keeps the hidden image difficult to detect, as shown by the high PSNR and correlation values for stego images. The algorithm must hide less information in images containing more smooth areas to keep avoiding detection. This indicates that hiding in such images would be a poor choice.

VI. FUTURE WORK

The presented algorithm may be modified easily to work with video where each frame is regarded as a single image. However, the modification should be made more efficient by taking advantage of video properties, which differ according to video content and format. For example, frames with less smooth contents could be detected and chosen for hiding information. Another reason for considering video format properties is their effect on video sensitivity to modification. For example, some video formats use the similarities and differences within frame sequences to perform compression. Hiding information in such videos may cause a detectable change in video size unless the hiding algorithm works around the compression method.

TABLE I. RESULTS FOR THE FACE TEST IMAGE

Cover Image	Correlation	PSNR (dB)	1 bit %	2 bit %	3 bit %	4 bit %
Valley	0.999988	59.174	36.9	32.0	27.1	4.0
Street	0.999986	57.731	56.3	23.0	15.3	5.4
Office	0.999994	62.701	84.6	11.3	2.9	1.2

TABLE II. AVERAGE RESULTS FOR 35 TEST IMAGES

Cover Image	Correlation	PSNR (dB)	1 bit %	2 bit %	3 bit %	4 bit %
Valley	0.999993	62.300	45.4	30.3	21.3	3.0
Street	0.999994	60.775	65.7	18.9	11.1	4.3
Office	0.999997	66.072	85.8	10.5	2.7	0.9

REFERENCES

[1] A. Al-Othmani, A. Abdul Manaf and A. Zeki, "A survey on steganography techniques in real time audio signals and evaluation," Int. J. Comput. Sci. Issues, vol. 9, no. 1, p. 3037, 2012.

[2] R. Amirtharajan, J. Qin and J. Rayappan, "Random image steganography and steganalysis: Present status and future direction," Inf. Technol. J., vol. 11, no. 5, pp. 566-576, 2012.

[3] G. Chhajed, K. Deshmukh and T. Kulkarni, "Review on binary image steganography and watermarking," Int. J. Comput. Sci. & Eng., vol. 3, no. 11, pp. 3645-3651, 2011.

[4] A. Hmood, H. Jalab, Z. Kasirun, B. Zaidan and A. Zaidan, "On the capacity and security of steganography approaches: An overview," J. Appl. Sci., vol. 10, no. 16, pp. 1825-1833, 2010.

[5] P. Jayaram, H. Ranganatha and H. Anupama, "Information hiding using audio steganography - A survey," Int. J. Multimedia Appl., vol. 3, no. 3, pp. 86-96, 2011.

[6] V. Reddy, A. Subramanyam and P. Reddy, "Implementation of LSB steganography and its evaluation for various file formats," Int. J. Adv. Networking Appl., vol. 2, no. 5, pp. 868-872, 2011.

[7] I. Shoukat, K. Abu Bakar and M. Iftikhar, "A survey about the latest trends and research isuues of cryptographic elements," Int. J. Comput. Sci. Issues, vol. 8, no. 3:2, pp. 140-149, 2011.

[8] S. Janakiraman, R. Amirtharajan, K. Thenmozhi and J. Rayappan, "Pixel forefinger for gray in color: A layer by layer stego," Inf. Technol. J., vol. 11, no. 1, pp. 9-19, 2012.

[9] A. Pradhan, D. Sharma and G. Swain, "Variable rate steganography in digital images using two, three and four neighbor pixels," Indian J. Comput. Sci. & Eng., pp. 457-463, 2012.

[10] M. Al-Husainy, "A new image steganography based on decimal-digits representation," Comput. & Inf. Sci., vol. 4, no. 6, pp. 38-47, 2011.

[11] O. Zanganeh and S. Ibrahim, "Adaptive image steganography based on optimal embedding and robust against Chi-square attack," Inf. Technol. J., vol. 10, no. 7, pp. 1285-1294, 2011.