VARIABLE-RATE STEGANOGRAPHY USING RGB STEGO-IMAGES

Ayman M. Abdalla, PhD

Dept. of Multimedia Systems, Al-Zaytoonah University, Amman, Jordan

Abstract

A new algorithm is presented for hiding information in the least significant bits of color images. The number of bits used for hiding changes according to pixel neighborhood information where less resemblance between a pixel and its neighbors leads to using more bits for hiding. Experimental results are presented to show that the algorithm generally hides information with a good capacity while avoiding detection.

Keywords: Image steganography, Information hiding, LSB method

Introduction

Steganography is a method of hiding a 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.

Surveys of different steganography techniques were presented in previous work, where secret information may be hidden in text, audio, image or video (Al-Othmani, et al., 2012), (Amirtharajan, et al., 2012), (Chhajed, et al., 2011), (Hmood, et al., 2010), (Jayaram, et al., 2011), (Reddy, et al., 2011). When an image is used for hiding information, it is called a stego image.

Hiding in the least significant bits (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 (Janakiraman, et al., 2012), (Pradhan, et al., 2012). 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 append coding information to the image with minimal or no change to the original pixels (Al-Husainy, 2011), (Zanganeh & Ibrahim, 2011).

In this paper, a new algorithm is presented to hide information in LSBs of image pixels. The algorithm uses a variable number of bits for each pixel, where the number of bits is chosen based on the amount of degradation they may cause to the pixel compared to its neighbors. Analysis showed effectiveness of the algorithm in minimizing degradation while maintaining substantial hiding capacity.

The Hiding Algorithm

This algorithm uses a variable number of LSBs from each pixel for hiding, where the number of bits chosen from each pixel color (red, green, and blue) is different. Images in other color formats may be converted to Red-Green-Blue (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 a pixel color and its neighbors is high, the pixel is located in a smooth area where change will be detected easily. Therefore, the number of bits used for hiding is chosen to be inversely proportional to the neighbors' average value of each pixel color.

The pixels used in hiding are those located in every line and every other column, 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 capacity. Each RGB color is treated separately. The hiding process starts with the Red matrix, followed by the Green, and then the Blue. The color value of each one of these pixels is compared to the average of the same-color values of its four neighbors: left, right, above, and below. This comparison measures the resemblance between the pixel and it neighbors 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. 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. Each examined entry is compared to the average (*ave*) of its four neighbors. If the

magnitude of the difference between the entry and *ave* is less than a given threshold (α), the minimum number of LSBs is used for hiding. In the implementation of this paper, α was set to 7. Otherwise, the floor of the logarithm (base 2) of this difference will be the number of LSBs (*numLSBs*) used, which is at least one bit larger than the minimum. The minimum value is changed by subtracting a small integer (ε) to make the minimum take the values 1, 2, or 3 bits.

An entry is used for hiding only if *ave* is at least twice as large as the largest possible value to be hidden in that entry. The actual hidden value is a number composed of *numLSBs* bits from the hidden message. If *ave* is larger than the original entry value in the stego image matrix, the new value of the entry will become ($ave - 2^{numLSBs} + b$); otherwise, it will become ($ave - 2^{numLSBs} - b$).

```
row = 2
while row ≤ n-2 and message is not finished
   col = 2 + (row MOD 2)
   while col ≤ m-2
      ave = (stegoC(row-1,col)+stegoC(row+1,col)+stegoC(row,col-1)+stegoC(row,col+1))/4
      dif = ave - stegoC(row,col)
      if |dif| ≤ α
         numLSBs = 3 - ε
      else
         numLSBs = ⌊log2(|dif|)⌋ - ε
      endif
      if ave ≥ 2^(numLSBs+1)
         b = number composed of next group of numLSBs bits from hidden message
         if dif > 0
            stegoC(row,col) = ave - 2^numLSBs + b
         else
            stegoC(row,col) = ave - 2^numLSBs - b
         endif
      endif
      col = col + 2
   endwhile
   row = row+1
endwhile
```

Fig. 1. Algorithm for hiding in one color matrix

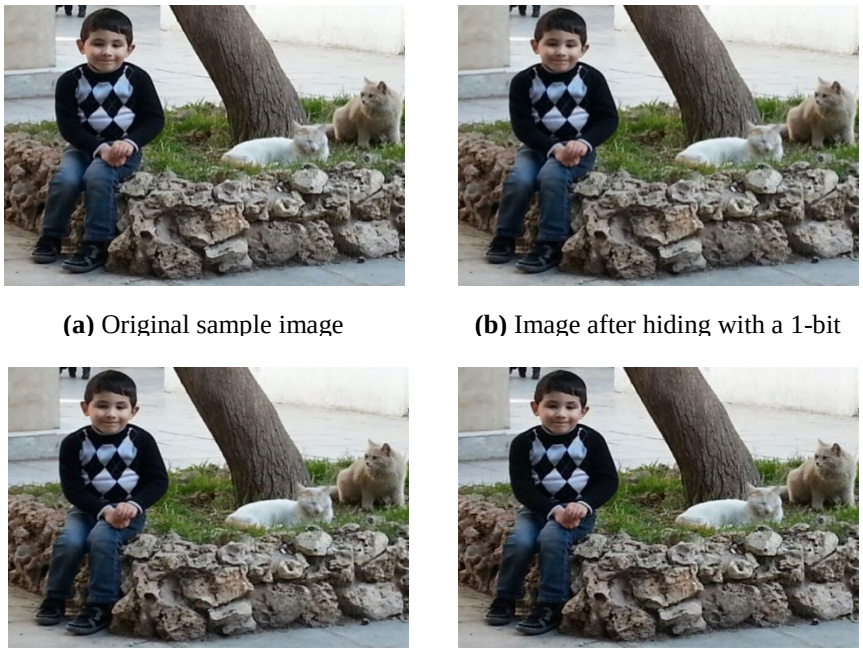
The extraction process searches each of the three color matrices (Red, Green, and Blue), going through all lines and every other column as in the hiding procedure. The number of bits used for hiding in an entry, *stegoC*(*row, col*) is also determined by examining *ave*; the average of the four neighbors as in the hiding process. If *ave* is larger than $2^{numLSBs+1}$, there is

a value hidden in the entry, which is $|stegoC(row, col) - ave + 2^{numLSBs}|$. All extracted hidden values are concatenated to form the original message.

Implementation and Analysis

The algorithm was applied to 50 different color images of different types, where the sizes of these images ranged from 10 to 3,578 kilobytes. A random secret message of sufficient length was used for hiding. The analysis of the results focus on two aspects: hiding capacity and difficulty to detect the message existence in the stego image. The capacity is computed as the ratio of the maximum hidden message size to the stego image size. The results were compared using different values for the minimum number of bits allowed for hiding. Recall that only non-adjacent pixels are used for hiding. These are approximately 50% of the pixels in the image.

Fig. 2 demonstrates the results for one sample image. It shows the original image and the stego images after hiding with a minimum of 1, 2, and 3 bits per pixel color. As seen in the figure, the difference between the original image and the stego images is not easily visible. When examining these four images closely, the stego image that used a minimum of 3 bits for hiding appears slightly darker than the other three do. The comparison becomes clearer when the following measurements are examined. The hiding capacity for this sample increased from



(a) Original sample image **(b)** Image after hiding with a 1-bit

(c) Image after hiding with a 2-bit **(d)** Image after hiding with a 3-bit

Fig. 2. Original sample image and stego images after hiding

6.42% to 12.45% and 17.87% when the minimum number of hiding bits was increased from 1 to 2 and 3, respectively. However, the peak signal-to-noise ratio (PSNR) decreased from 35.80 to 33.95 and 30.59 decibels (dB) and the correlation decreased from 0.9977 to 0.9968 and 0.9938 for these respective values for the minimum numbers of hiding bits.

The average results for all 50 test images are shown in Table 1. The average correlation value was taken for the absolute values of correlation for all images, where each of the original images was compared to its stego image to obtain the individual correlation values. The overall high values of PSNR and correlation indicate big resemblance between the original images and their stego images, and consequently, less hidden-message detection ability.

Table 1. Average results for 50 test images

Min Hide Bits	Capacity	PSNR (dB)	Correlation
1	6.54%	35.23	0.9938
2	12.27%	32.97	0.9921
3	17.02%	29.70	0.9877

As seen in Table 1, the average hiding capacity for all 50 images increased when the minimum number of hiding bits was increased where the PSNR and correlation values decreased. The capacity increase when using a minimum of 2 hiding bits rather than 1 was 5.73 percentage points, which is an increase of 87.6% in the average capacity value. The cost of this increase was lowering the PSNR and correlation values by 6.4% and 0.17%, respectively. When the minimum number of hiding bits was increased from 2 to 3, the capacity increased by 38.7% where PSNR and correlation values decreased by 9.9% and 0.44%, respectively. This is a lower increase in capacity with a larger decrease in PSNR and correlation values compared to the changes that resulted when going from 1 to 2 hiding bits. This indicates that using a minimum of 2 bits for hiding provided a relatively better tradeoff between hiding capacity and stego image quality than using a minimum of 1 or 3 bits.

Conclusion

The new algorithm presented in this paper uses a variable number of LSBs from each color of each considered pixel for hiding information, where approximately 50% of all pixels are considered for hiding. The actual number of hiding bits is inversely proportional to the pixel's resemblance to its neighbors. Test results showed that the algorithm has a good capacity for hiding information while keeping the hidden information difficult to detect.

References:

A. Al-Othmani, A. Abdul Manaf and A. Zeki, “A survey on steganography techniques in real time audio signals and evaluation,” International Journal of Computer Science Issues, vol. 9, no. 1, p. 3037, 2012.

R. Amirtharajan, J. Qin and J. Rayappan, “Random image steganography and steganalysis: Present status and future direction,” Information Technology Journal, vol. 11, no. 5, pp. 566-576, 2012.

G. Chhajed, K. Deshmukh and T. Kulkarni, “Review on binary image steganography and watermarking,” International Journal on Computer Science & Engineering, vol. 3, no. 11, pp. 3645-3651, 2011.

A. Hmood, H. Jalab, Z. Kasirun, B. Zaidan and A. Zaidan, “On the capacity and security of steganography approaches: An overview,” Journal of Applied Sciences, vol. 10, no. 16, pp. 1825-1833, 2010.

P. Jayaram, H. Ranganatha and H. Anupama, “Information hiding using audio steganography - A survey,” International Journal of Multimedia & Its Applications, vol. 3, no. 3, pp. 86-96, 2011.

V. Reddy, A. Subramanyam and P. Reddy, “Implementation of LSB steganography and its evaluation for various file formats,” International Journal of Advanced Networking & Applications, vol. 2, no. 5, pp. 868-872, 2011.

S. Janakiraman, R. Amirtharajan, K. Thenmozhi and J. Rayappan, “Pixel forefinger for gray in color: A layer by layer stego,” Information Technology Journal, vol. 11, no. 1, pp. 9-19, 2012.

A. Pradhan, D. Sharma and G. Swain, “Variable rate steganography in digital images using two, three and four neighbor pixels,” Indian Journal of Computer Science & Engineering, pp. 457-463, 2012.

M. Al-Husainy, “A new image steganography based on decimal-digits representation,” Computer & Information Science, vol. 4, no. 6, pp. 38-47, 2011.

O. Zanganeh and S. Ibrahim, “Adaptive image steganography based on optimal embedding and robust against Chi-square attack,” Information Technology Journal, vol. 10, no. 7, pp. 1285-1294, 2011.