Multibit Assignment Steganography in Palette Images

Xinpeng Zhang, Shuozhong Wang, and Zhenyu Zhou

***Abstract*—This letter proposes a novel multibit assignment steganography for palette images, in which each gregarious color that possesses close neighboring colors in the palette is exploited to represent several secret bits. A method for producing a usable multibit assignment is presented. For any pixel with a gregarious color, a data-hider can always find a suitable neighbor of the original color that corresponds to a prefix of secret bit-sequence and then replaces the original color with the found neighboring color. Experiment results show that the proposed steganography provides a good payload-distortion performance.**

***Index Terms*—Multibit assignment, palette image, steganography.**

I. Introduction

THE purpose of steganalysis is to detect the presence of secret message embedded in a carrier signal using steganographic techniques [1]. To ensure security, the data-hider always attempts to keep distortion due to data embedding as low as possible. Various types of digital contents can be used as the covers, among which palette images are popular since they are widely available and convenient to distribute over the Internet. Palette image uses a few, generally no more than 256, colors to provide acceptable visual quality, and each pixel possesses an index value mapped to a displayed color according to a palette containing all colors in the image.

Gifshuffle is a program used to conceal messages in GIF images, available online [2]. By reordering a palette that possesses N colors, $\log_2(N!)$ bits of additional information can be inserted since any list of N items can be arranged in $N!$ different ways. Although this method does not change the appearance of the image, a randomly ordered palette provides steganalysts a useful clue, and rearranging the palette can easily remove the embedded information. In [3], the pixels of a cover palette image are used to carry the secret message, and the original palette is also modified for lowering distortion. If the distortion caused by data-embedding when removing an original color in the palette exceeds that caused by generation of a new color, the original color will be replaced with the new one. This method can provide a high visual imperceptibility. However, since the added colors are always very close to the existing ones, a steganalyst may detect the presence of hidden message from the palette abnormality.

Using another type of steganographic methods [4], [5], the palette is kept unchanged, and only the pixels are modified to accommodate the secret data. In this way, all colors in the palette are divided into two subsets representing, respectively, the secret bits 0 and 1. For a pixel into which one secret bit is embedded, no modification is needed if the original color belongs to the corresponding subset; otherwise, the closest color in another subset is chosen to replace the original color. In [4], assignment of colors to the subsets is done according to the parity of the sum of red, green, and blue components. In [5], a smarter optimal parity assignment (OPA) method is introduced, in which a color in the palette and its closest neighbor must belong to two different subsets. Thus, the original color of any pixel is either kept unchanged or modified into its closest neighbor. The induced distortion is therefore very small.

Actually, a palette may have several colors with small and similar distances to a certain color. So, when altering the original color of a pixel for data embedding, a data-hider can select a suitable candidate among a number of neighbor-colors to replace the original color. This implies that each pixel can be exploited to carry more than one secret bit. In view of this, here we propose a multibit assignment steganographic scheme, which causes lower distortion or achieves a larger payload than the OPA method.

II. Framework of Multibit Assignment Steganography

In the multibit assignment steganography, the original palette is not changed and the colors in the palette are assigned to carry several bits. This section describes the general principle of usable multibit assignment and presents a data embedding procedure. The method for producing a usable multibit assignment will be described in the next section.

Definition 1: If the distance between two colors $c_1 = [r_1, g_1, b_1]$ and $c_2 = [r_2, g_2, b_2]$ in a palette

$$d = \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2} \quad (1)$$

is less than a given threshold D, we say that $c_1(c_2)$ is a neighbor color of $c_2(c_1)$.

Definition 2: If a color possesses at least one neighbor color, we say it is a gregarious color, and we call a set containing a gregarious color and its all neighbors as a neighborhood set.

Manuscript received March 27, 2008; revised May 12, 2008. This work was supported in part by the National Natural Science Foundation of China under Grants 60502039 and 60773079 and in part by the High-Tech Research and Development Program of China under Grant 2007AA01Z477. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Athanassios Skodras.

The authors are with School of Communication and Information Engineering, Shanghai University, Shanghai 200072, China (e-mail: xzhang@shu.edu.cn; xzhang.shu@gmail.com; shuowang@shu.edu.cn; rinick@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2008.2001117

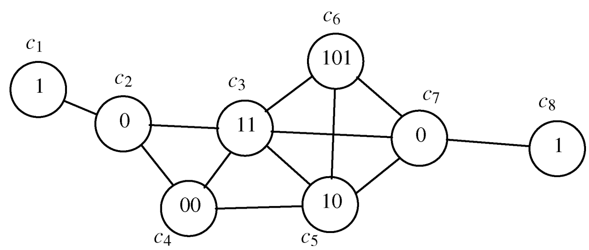

Fig. 1. Example of usable multibit assignment on eight colors.

Definition 3: After the gregarious colors in a palette are assigned to carry different bit-chips with various lengths, if, for any gregarious color and any bit sequence with an indefinite length, there are always at least one bit-chip mapped to a member of the gregarious color's neighborhood set that is a prefix of the given bit sequence, we say the multibit assignment is usable.

For example, Fig. 1 gives a usable multibit assignment on a palette with eight colors. Here, an edge connecting two different colors indicates a neighborhood relationship between them, and all the colors are gregarious. For the color c_1, its neighborhood set is $\{c_1, c_2\}$ with the corresponding bit-chips 1 and 0, and any bit sequence must begin with 1 or 0. For the color c_4, the neighborhood set is $\{c_2, c_3, c_4, c_5\}$ with the corresponding bit-chips 0, 11, 00, and 10, and any bit sequence with an indefinite length must begin with 0, 11, or 10. If the first two bits of a bit sequence are 00, both 0 and 00 are the prefixes. In fact, for any color in the palette, we can always find a bit-chip in its neighborhood set that is a prefix of any given bit sequence. That means the multibit assignment in Fig. 1 is usable.

The data embedding procedure is as follows.

1) According to a given threshold D and a secret key, a data-hider produces a usable multibit assignment. A usable multibit assignment method will be described in the next section.
2) Convert the secret message into a binary sequence, and permute all cover pixels with gregarious colors as a pixel-sequence in a pseudo-random way derived from the secret key.
3) If the bit-chip corresponding to the color of the first pixel in the pixel-sequence is a prefix of secret bit sequence, no modification is needed. Otherwise, find a neighbor of the color that maps a bit-chip coincident with a prefix of secret bit sequence, and replace the original color with the neighbor color. Because the multibit assignment is usable, the data-hider can always find at least one suitable neighbor meeting this condition. If there are several suitable neighbors, the data-hider randomly selects one as the new color. In this case, the prefix of secret bit sequence is carried by this pixel. Note that the new color is still gregarious. Then, remove the prefix and the pixel from the secret bit sequence and pixel sequence, respectively, and repeat this step to embed the remaining bits into the rest pixels until all pixels are used.
4) By inverse permutation, the data-hider uses the pixels carrying the secret bit-chips to form a stego-image.

Data-extraction is simple. With the parameter D and the secret key, a receiver first obtains the same multibit assignment result and the permuted sequence of pixels with gregarious colors. Then, he can concatenate all bit-chips corresponding to the pixel-colors to obtain the secret message.

Assume multibit assignment as shown in Fig. 1 is used. The permuted pixel-sequence is $[c_2c_8c_4c_5c_1c_7c_3c_4c_6c_3]$, and the secret message-sequence is [0011011010101100]. Using the proposed data-embedding procedure, the stego-pixel-sequence will be $[c_2c_7c_3c_7c_1c_6c_2c_5c_3c_4]$. This way, ten pixels are used to carry 16 secret bits.

III. Multibit Assignment Method

In this section, a layer-by-layer method for producing a usable multibit assignment is presented. Within each layer, we assign one suitable bit to some gregarious colors. Then, the bits assigned to a color in different layers are orderly concatenated to form a bit-chip carried by this color.

Steps of the first layer assignment are as follows.

1) Calculate the distances between all pairs of colors, and compare it with the threshold D. If the distance $d_{i,j}$ between c_i and c_j is less than D, they are neighbors of each other, and we assign an edge $E_{i,j}$ between them. Set $C = \varnothing$. Iteratively repeat the next step until C contains all gregarious colors.
2) Pseudo-randomly choose an edge $E_{i,j}$ using a key such that either $c_i \notin C$ or $c_j \notin C$. No such $E_{i,j}$ can be found if C already contains all gregarious colors. If neither c_i nor c_j belongs to C, pseudo-randomly assign a bit 0 and 1 to c_i and c_j, respectively. In case $c_i \notin C$ and $c_j \in C$, a binary value different from the bit corresponding to c_j is assigned to c_i. Update $C = C \cup \{c_i\} \cup \{c_j\}$.

After the first layer assignment, each gregarious color is mapped to a bit "0" or "1". Also, for any gregarious color, there must be at least one neighbor corresponding to a reverse bit. That implies the first layer assignment is usable.

Assuming a usable m-layer assignment has been produced, meaning that the maximum number of bits having been assigned to a gregarious color is m, we then perform the $(m+1)$th layer assignment.

1) Collect all gregarious colors that possess m assigned bits, and divide them into 2^m subsets according to the values of their assigned bits. In other words, colors in a subset are used to represent the same type of secret bit-chip with a length m. For each non-empty subset, the following steps are taken to assign one more bit to some its gregarious colors.
2) Considering each gregarious color in a certain subset, if the color possesses at least one neighbor in the same subset, we say it is m-layer-gregarious. Set $C = \varnothing$. Iteratively repeat the next step until C contains all m-layer-gregarious colors in the subset.
3) According to the key, pseudo-randomly choose an edge $E_{i,j}$ where both c_i and c_j belong to the subset, and either $c_i \notin C$ or $c_j \notin C$. No such $E_{i,j}$ can be found if C already contains all m-layer-gregarious colors. If neither c_i nor c_j belongs to C, pseudo-randomly assign a new bit 0 and 1 to c_i and c_j, respectively. In case $c_i \notin C$ and $c_j \in C$, a

binary value that is reverse of the new bit of c_j is assigned to c_i so that the new bits of c_i and c_j are different. Update $C = C \cup \{c_i\} \cup \{c_j\}$.

4) After Steps 2 and 3, each m-layer-gregarious color in the subset gets its new bit. Denote a set containing the neighbors of the m-layer-gregarious colors except the m-layer-gregarious colors themselves as a check-set. For each element in the check-set, if all its neighbors in the subset are m-layer-gregarious and all the bits newly assigned to them are identical, we pseudo-randomly select one color from them, and cancel the bit newly assigned to the selected color.
5) The remaining bits are regarded as the actual $(m+1)$th bits assigned to their corresponding colors.

Note that the purpose of Step 4 is to ensure the $(m+1)$th layer assignment is still usable. If any actual assigned bit has not been produced in a certain layer, terminate the multibit assignment procedure; otherwise, continue the next layer assignment.

For example, with the neighborhood relationship of eight colors as shown in Fig. 1, in the first layer, the colors c_2, c_4 ,and c_7 are mapped to "0", while the colors c_1, c_3, c_5, c_6, and c_8 are mapped to "1". Then, $\{c_2, c_4, c_7\}$ and $\{c_1, c_3, c_5, c_6, c_8\}$ are two subsets in the second layer. For the first subset, c_2 and c_4 are 1-layer-gregarious, and assign new bits "1" and "0" to them, respectively. Here, the check-set is $\{c_1, c_3, c_5\}$. For c_3 and c_5, cancellation of any new bit is not needed, while, for c_1, the bit "1" assigned to c_2 should be cancelled since c_1 has only one 1-layer-gregarious neighbor c_2. For the second subset $\{c_1, c_3, c_5, c_6, c_8\}$, c_3, c_5, and c_6 are 1-layer-gregarious, and assign new bits "1", "0", and "0" to them, respectively. When considering the elements in check-set $\{c_2, c_4, c_7\}$, we do not cancel any new bit. In the third layer, there are three non-empty subsets: $\{c_4\}$ corresponding to two bits "00", $\{c_5, c_6\}$ corresponding to two bits "10", and $\{c_3\}$ corresponding to two bits "11". Since both the subset $\{c_4\}$ and $\{c_3\}$ possess only one element, the colors c_4 and c_3 are not 2-layer-gregarious so that no new bit is produced. For the subset $\{c_5, c_6\}$, both c_5 and c_6 are 2-layer-gregarious, and new bits "0" and "1" are assigned to them, respectively. Then, the check-set is $\{c_3, c_4, c_7\}$. When considering the element c_4, since c_5 is its only neighbor in the subset, we cancel the bit "0" assigned to c_5. In the fourth layer, since the only subset $\{c_6\}$ contains just one color, it does not produce any assigned bit and the multibit assignment procedure terminates. Fig. 1 shows the final result of a usable assignment status.

IV. Experimental Results

In the experiment, two 256-color palette images Lena and Baboon, both sized 512 × 512, were used as covers. Using a threshold $D = 12$, 3.7×10^5 secret bits were embedded into Lena with PSNR = 36.19 dB and MSE = 15.6. For Baboon using $D = 20$, the number of embedded bits was 2.6×10^5, PSNR = 32.27 dB and MSE = 38.5. The two stego-images are shown in Fig. 2. Since Lena has a narrower color range than Baboon, it can carry a larger payload with lower distortion. When using another cartoon image Fight sized 352 × 305 as the cover and using $D = 16$, a total of 1.2×10^5 secret bits was concealed with PSNR = 33.20 dB and MSE = 31.1. The original and stego Fight are shown in Fig. 3.

(a) (b)

Fig. 2. Stego-images: (a) Lena and (b) Baboon.

(a) (b)

Fig. 3. (a) Original cartoon Fight and (b) its stego-version.

Different threshold values result in different payloads and distortion levels as shown by the payload-distortion curves presented in Fig. 4. The abscissa represents the number of embedded secret bits per cover pixel, i.e., the embedding rate, and the ordinate is PSNR. Performance of the OPA method is also given in the figure, which is clearly inferior to the proposed multibit assignment scheme. The performance comparison is also made on 100 cover images, including 50 cartoons (drawings) and 50 landscapes. The results show that the mean ratio between MSE due to the proposed method and that due to OPA with the same payload is 0.61, indicating a lower PSNR for the proposed method with an average gain of 2.14 dB. Conversely, when keeping the same level of distortion, data capacity of the proposed method is 1.62 times that of OPA on average.

V. Conclusion and Discussion

The steganographic technique proposed in this letter uses each gregarious color to represent a secret chip of several bits. For any pixel with a gregarious color, one can always find a suitable color in the original color's neighborhood set and replace the original color with it to hide at least one secret bit. This way, payload is increased, or distortion is decreased when comparing with OPA method.

On the other hand, introducing a secret key in the multibit assignment procedure enhances security of the hidden information. In the OPA method [5], the original color of a pixel

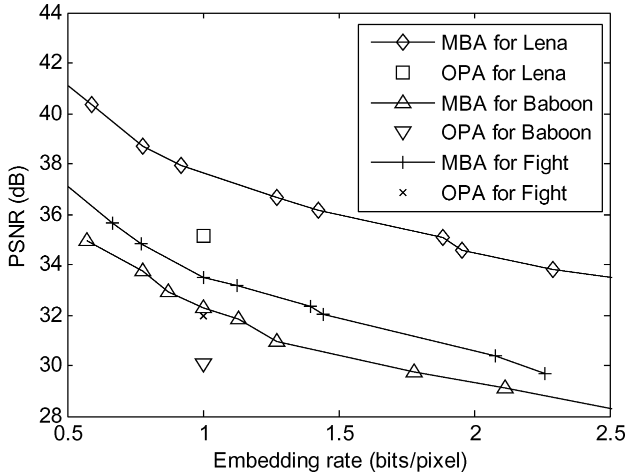

Fig. 4. Performance comparison between the proposed MBA method and OPA.

is either kept unchanged or modified into its closest neighbor. Because the embedding rule is also known to the steganalyst, having found the mapping relationship between the colors and their closest neighbors, one can attempt to recover the original histogram in a reverse way of the data embedding. Since excessive operations can cause negative values in the recovered histogram, the steganalyst is able to detect the presence of secret message and estimate the quantity of embedded bits [6]. In contrast, since the status of multibit assignment is determined by the secret key in the proposed scheme, it is difficult to guess the modification relationship between the gregarious colors. That means the presence of secret message cannot be detected by the attempt of recovering the original histogram.

In some steganographic techniques based on the human visual system (HVS), busy areas containing texture or edge contents are used to conceal more secret information since these areas can tolerate more changes [7], [8]. Clearly, it will be beneficial to take into account the HVS in the multibit assignment mechanism. For example, the value of D may be adaptively selected in different regions according to the magnitude of local fluctuation, giving a better visual imperceptibility. In the future studies, we will consider steganographic schemes that incorporate HVS characteristics in the multibit assignment mechanism to further improve the performance.

REFERENCES

[1] H. Wang and S. Wang, “Cyber warfare: Steganography vs. steganalysis,” *Commun. ACM*, vol. 47, no. 10, pp. 76–82, 2004.

[2] [Online]. Available: http://www.darkside.com.au/gifshuffle/.

[3] M.-Y. Wu, Y.-K. Ho, and J.-H. Lee, “An iterative method of palette-based image steganography,” *Pattern Recognit. Lett.*, vol. 25, pp. 301–309, 2004.

[4] J. Fridrich, “A new steganographic method for palette-based images,” in *Proc. IS&T PICS Conf.*, Savannah, GA, Apr. 1998, pp. 285–289.

[5] J. Fridrich and R. Du, “Secure steganographic methods for palette images,” in *The 3rd Information Hiding Workshop, Lecture Notes in Computer Science*, 2000, vol. 1768, pp. 47–60.

[6] X. Zhang and S. Wang, “Analysis of parity assignment steganography in palette images,” in *Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Artificial Intelligence*, 2005, vol. 3683, pp. 1025–1031.

[7] D. C. Wu and W. H. Tsai, “A steganographic method for images by pixel-value differencing,” *Pattern Recognit. Lett.*, vol. 24, pp. 1613–1626, 2003.

[8] X. Zhang and S. Wang, “Steganography using multiple-base notational system and human vision sensitivity,” *IEEE Signal Process. Lett.*, vol. 12, no. 1, pp. 67–70, Jan. 2005.