

Topological Ordered Color Table for BPCS Steganography Using Indexed Color Images

ROBERT OUELLETTE,[†] HIDEKI NODA,[†] MICHIHARU NIIMI[†]
and EIJI KAWAGUCHI[†]

A new method is presented for applying bit-plane complexity segmentation (BPCS) steganography to 8-bit color images. By reorganizing the indices of 8-bit indexed color files using Kohonen's self-organizing neural network, we have created a topological ordered color table in which nearby indices have similar color components. Consequently, we have been able to embed data whose sizes are 20–40% of the dummy files, with very little degradation in the dummy image.

1. Introduction

Steganography is the practice of hiding or camouflaging information within a dummy container. The main requirement in effective steganography is that the dummy container should give the appearance of “innocence” without alerting outside observers to the fact that additional information is hidden within it. Bit-plane decomposition has been commonly used in the past steganography with digital image and sound data employed as the dummy containers. A simple approach that is frequently used is to replace the binary data in the least significant bit-plane with secret binary data²⁾.

In a previous work, we presented a sophisticated steganography method, called bit-plane complexity segmentation (BPCS) steganography, which makes use of the bit-plane decomposition and the characteristics of the human vision system³⁾. Noting that humans cannot perceive any shape information in a very complicated binary pattern, we can replace noise-like regions in the bit-planes of the dummy image with secret data without deteriorating the image quality. The BPCS steganography makes it feasible to embed surprisingly large amounts of information, with a size of up to 50% of that of true color dummy image⁴⁾. Until recently, however, it could only be used with true color (24-bit) images as color dummy images. This limitation was a result of the dependency on the format in which the data was stored. True color images use a direct relationship between the data being stored and the output, while

other file types use indexing and/or compression schemes to store the data. In this paper, we have focused on indexed 8-bit color images as containers, primarily because of their popularity and simplicity in format.

The indices in indexed color images have no specified ordering. This causes a fatal problem in BPCS steganography. An example is shown in Fig. 2 (b) (see Section 4). Random binary data was embedded into the least significant eighth bit-plane of a dummy image shown in Fig. 2 (a) with the complexity threshold $\alpha_0 = 35/112$ (see Section 2). Although the change of each pixel's value in this embedding example is only one at most, the color is heavily changed because of the lack of ordering in the color table. Therefore, in order for BPCS steganography to be effective for indexed images, pixels that are similar in color should have similar index values. Some ordering methods, such as the ordering of color table entries according to the magnitudes of their color vectors or their luminance values, are described in Ref. 2). However, it is clear that such methods do not work for entries whose color vector components are completely different but have similar magnitudes or luminance values. Instead, we need a more sophisticated ordering which represents an underlying topology of the distribution of entries in the three-dimensional color

For example, an n -bit image can be decomposed into a set of n binary images by bit-slicing operations¹⁾. 8-bit image data values are actually pointers to entries in a color table. The table entry to which the pixel value points holds the 8-bit red, green, and blue values of the pixel. Because the data values are 8-bit, the index table is limited to a maximum of 256 entries.

[†] Kyushu Institute of Technology

space. A topological ordered color table can be created by using Kohonen's self-organizing neural network⁵⁾. Such color tables result in indexed color images prime for BPCS steganography.

2. Overview of BPCS Steganography

It is necessary to somehow relate noise-like regions to non-noise-like regions (which we call informative regions) in order to describe one in terms of the other. The two can be related by defining a complexity measure, which is a property common to both. The definition for complexity currently accepted in BPCS steganography is the length of binary image (black-and-white) border. The complexity α for an $m \times m$ size binary image is defined as

$$\alpha = \frac{k}{2m(m-1)}, \quad 0 \leq \alpha \leq 1, \quad (1)$$

where k is the total length of the black-and-white border in the image and $2m(m-1)$ is the maximum possible border length obtained from an $m \times m$ checkerboard pattern.

A typical procedure for information hiding in BPCS steganography is as follows.

- (1) Segment each bit-plane of a dummy image into 8×8 size blocks. Then classify these blocks into informative and noise-like blocks using a threshold of the complexity α_0 .
- (2) Segment a secret file into a series of blocks each containing 8 bytes of data. These blocks (which we call secret blocks) are regarded as 8×8 binary images.
- (3) If a secret block is less complex than the threshold α_0 , conjugate it to make it more complex.
- (4) Embed each secret block into the noise-like blocks of the bit-plane. If the block is conjugated, then record this fact in a conjugation map.
- (5) Also embed the conjugation map in the same way as the secret blocks.

3. Topological Ordered Color Table

The BPCS algorithm is based on the fact that noise-like data at the bit-plane level can be replaced with other noise-like secret data with no visibly perceptible change. However, this fea-

ture only works in images where pixels which have the same or similar values will respectively have the same or similar output colors. Unfortunately, 8-bit color images are not structured in such a way.

Thus, in order to apply traditional BPCS steganography to 8-bit color images, it is necessary to overcome the unordered nature of the data index table. One solution is to reorganize the index table using the self-organizing feature map (SOFM) by Kohonen⁵⁾ so that nearby indices have entries that are similar to each other. Thus, regions of pixels that are similar in color will have similar index values. Consequently, the newly ordered index table should be acceptable as a container for BPCS steganography.

The one-dimensional SOFM used here consists of 256 neurons, where the number 256 corresponds to the number of indices. Let the three-dimensional weight vector of the i -th neuron \mathbf{w}_i , $i = 1, \dots, 256$ and a three-dimensional input vector \mathbf{x} , whose components consists of three color components. The learning rule is summarized as follows⁶⁾:

- (1) **Initialization.** Choose random values for the initial weight vectors $\mathbf{w}_i(0)$ s.
- (2) **Input Sampling.** Draw a pixel randomly from a given indexed image. Use three color values associated with the pixel value (index) as the input vector $\mathbf{x}(t)$ at time t .
- (3) **Similarity Matching.** Find the best-matching (winning) neuron $c(t)$ at time t , using the minimum-Euclidean-distance criterion

$$c(t) = \arg \max_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|. \quad (2)$$

- (4) **Updating.** Adjust the weight vectors by using the update formula

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + \eta(t)(\mathbf{x}(t) - \mathbf{w}_i(t)) & i \in N_{c(t)}, \\ \mathbf{w}_i(t) & \text{otherwise,} \end{cases} \quad (3)$$

where $\eta(t)$ is the learning-rate parameter and $N_{c(t)}$ is the neighborhood function centered on the winning neuron $c(t)$.

- (5) **Continuation.** Continue with step (2) until no noticeable changes in the feature map are observed.

Three real components of the finally obtained weight vector \mathbf{w}_i are converted into 8-bit integer values and then utilized for the i -th index. In the following experiment, we used the values

The process called conjugation is the exclusive OR operation with a checkerboard pattern. The relation $\alpha^* = 1 - \alpha$ holds true³⁾, where α and α^* are the complexity of a given image and that of the conjugated image, respectively.

suggested in Ref. 6) for the parameters $\eta(t)$ and $N_{c(t)}$ in Eq. (3).

4. Experimental Results

Figure 1 (a) shows the original ordering of the index of the 8-bit color image “girl” (256×256 size) shown in Fig. 2 (a). The lines connect consecutive entries in the index table. Clearly, there is no obvious ordering. The line in Fig. 1 (b) shows the index after it has been

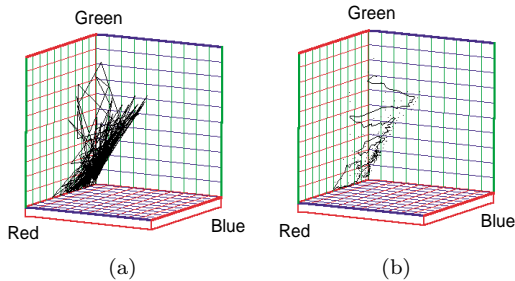


Fig. 1 Orderings of indexed color tables: (a) before, and (b) after reorganization.

reorganized by the SOFM algorithm. The index now has a nice topological order and, as is shown by its position against the original index data (the dots), it is still representative of the original index space.

In Fig. 2 (c), which shows the image with a reorganized index table, the quality is as good as in Fig. 2 (a). Using this image as a dummy container, data embedding experiments were carried out. A series of random binary values was used as secret data. Figures 2 (d), 2 (e) and 2 (f) show the embedded images, where 20%, 30% and 40% of the container size are embedded with the complexity threshold $\alpha_0 = 44/112, 35/112,$ and $18/112,$ respectively.

5. Conclusions

BPCS steganography has a high potential for use in Internet and data security applications. However, to gain widespread acceptance, it must be expanded to allow not only 24-bit true-color images and 8-bit monochrome im-

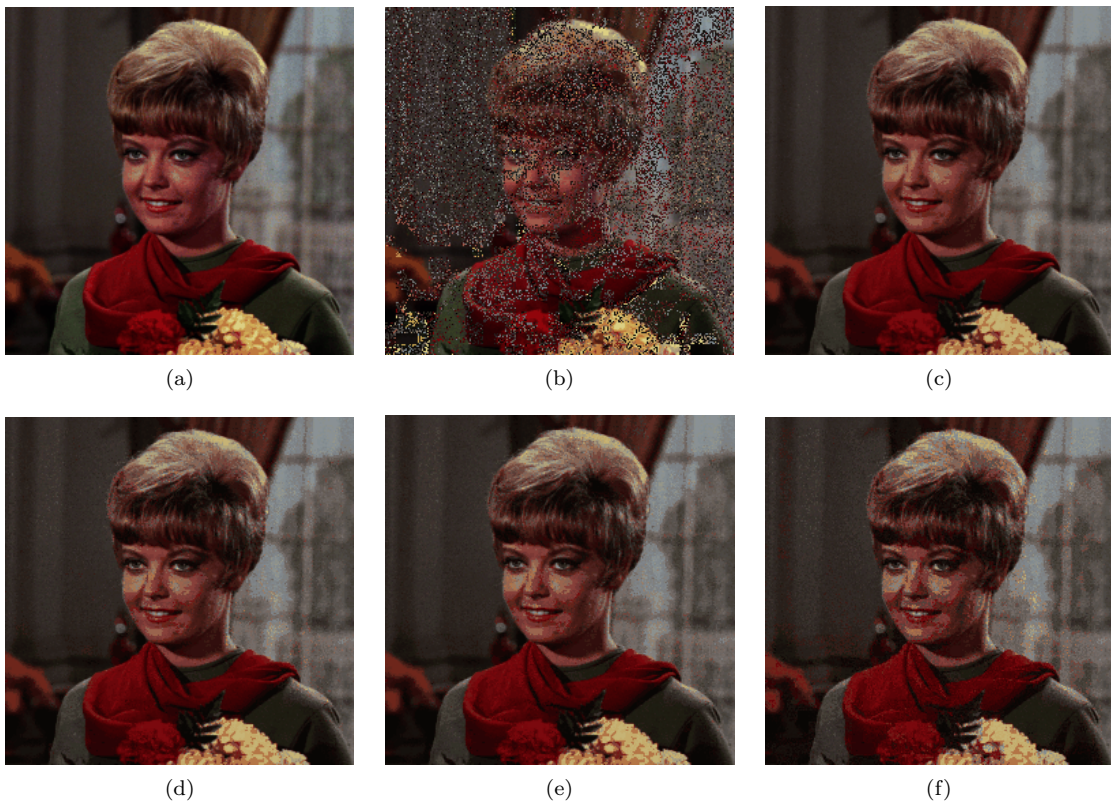


Fig. 2 Experimental results: (a) original indexed color image, (b) 11% embedded into Fig. (a), (c) image with reorganized index table, (d) 20% embedded, (e) 30% embedded, and (f) 40% embedded into Fig. (c).

ages, but also indexed and compressed color images which are more commonly used. In this paper, we have used Kohonen's self-organizing neural network to reorganize the indices of 8-bit indexed images, allowing BPCS steganography to extend its reach into indexed images. The percentage of embedded data in images can reach 20–40% with almost no or little noticeable change in the dummy container.

Another very important issue is to develop BPCS steganography to allow lossy compressed images to be used as dummy containers. In fact we are now developing a method which can integrate wavelet-based lossy compression with BPCS steganography⁷).

References

- 1) Jain, A.K.: *Fundamentals of Digital Image Processing*, Prentice Hall (1989).
- 2) Katzenbeisser, S. and Petitcolas, F.A.P.: *Information Hiding Techniques for Steganography and Digital Watermarking*, Artech House (2000).
- 3) Niimi, M., Noda, H. and Kawaguchi, E.: A Steganography Based on Region Segmentation by Using Complexity Measure, *Trans. IEICE*, Vol.J81-D-II, No.6, pp.1132–1140 (1998).
- 4) Kawaguchi, E. and Eason, R.O.: Principle and Applications of BPCS-Steganography, *Proc. SPIE*, Vol.3528, pp.464–473 (1998).
- 5) Kohonen, T.: *Self-Organization and Associative Memory*, Springer-Verlag (1989).
- 6) Haykin, S.: *Neural Networks, A Comprehensive Foundation*, pp.412–413, Macmillan College Publishing (1994).
- 7) Noda, H., Shirazi, M.N., Spaulding, J. and Kawaguchi, E.: Steganography Using Wavelet Compressed Images (submitted for Electronic Imaging 2001, San Jose).

(Received July 3, 2000)

(Accepted October 6, 2000)