BPCS-Steganography Experimental Program site:
http://www.datahide.org/BPCSe/QtechHV-program-e.html

Principle and applications of BPCS-Steganography

Eiji Kawaguchi* and Richard O. Eason**

* Kyushu Institute of Technology, Kitakyushu, Japan
** University of Maine, Orono, Maine 04469-5708

ABSTRACT

Steganography is a technique to hide secret information in some other data (we call it a vessel) without leaving any apparent evidence of data alteration. All of the traditional steganographic techniques have limited information-hiding capacity. They can hide only 10% (or less) of the data amounts of the vessel. This is because the principle of those techniques was either to replace a special part of the frequency components of the vessel image, or to replace all the least significant bits of a multi-valued image with the secret information.

Our new steganography uses an image as the vessel data, and we embed secret information in the bit-planes of the vessel. This technique makes use of the characteristics of the human vision system whereby a human cannot perceive any shape information in a very complicated binary pattern. We can replace all of the "noise-like" regions in the bit-planes of the vessel image with secret data without deteriorating the image quality. We termed our steganography "BPCS-Steganography," which stands for Bit-Plane Complexity Segmentation Steganography.

We made an experimental system to investigate this technique in depth. The merits of BPCS-Steganography found by the experiments are as follows.

1. The information hiding capacity of a true color image is around 50%.
2. A sharpening operation on the dummy image increases the embedding capacity quite a bit.
3. Canonical Gray coded bit planes are more suitable for BPCS-Steganography than the standard binary bit planes.
4. Randomization of the secret data by a compression operation makes the embedded data more intangible.
5. Customization of a BPCS-Steganography program for each user is easy. It further protects against eavesdropping on the embedded information.

Keywords: steganography, data hiding, information hiding, BPCS, digital picture envelope, vessel image, dummy image, encryption, compression, bit plane

1. INTRODUCTION

Internet communication has become an integral part of the infrastructure of today's world. The information communicated comes in numerous forms and is used in many applications. In a large number of these applications, it is desired that the communication be done in secrete. Such secret communication ranges from the obvious cases of bank transfers, corporate communications, and credit card purchases, on down to a large percentage of everyday email. With email, many people wrongly assume that their communication is safe because it is just a small piece of an enormous amount of data being sent worldwide. After all, who is going to see it? But in reality, the Internet is not a secure medium, and there are programs "out there" which just sit and watch messages go by for interesting information.

Encryption provides an obvious approach to information security, and encryption programs are readily available. However, encryption clearly marks a message as containing "interesting" information, and the encrypted message becomes subject to attack. Furthermore, in many cases it is desirable to send information without anyone even noticing that information has been sent.

Steganography presents another approach to information security. In steganography, data is hidden inside a vessel or container that looks like it contains only something else. A variety of vessels are possible, such as digital images, sound clips, and even executable files. In recent years, several steganographic programs have been posted on Internet home pages. Most of them use image data for the container of the secret information. Some of them use the least significant bits of the image data to hide the data. Other programs embed the secret information in a specific band of the spatial frequency component of the carrier. Some other programs make use of the sampling error in image digitization. However, all those steganographic techniques are limited in terms of information hiding capacity. They can embed only 5-15 % of the vessel image at the best. Therefore, current steganography is more oriented to water marking of computer data than to secret person-person communication applications.

We have invented a new technique to hide secret information in a color image. This is not based on a programming technique, but is based on the property of human vision system. Its information hiding capacity can be as large as 50% of the original image data. This could open new applications for steganography leading to a more secure Internet communication age.

Digital images are categorized as either binary (black-and-white) or multi-valued pictures despite their actual color. We can decompose an n-bit image into a set of n binary images by bit-slicing operations [1][2]. Therefore, binary image analysis is essential to all digital image processing. Bit slicing is not necessarily the best in the Pure-Binary Coding system (PBC), but in some cases the Canonical Gray Coding system (CGC) is much better [3].

2. THE COMPLEXITY OF BINARY IMAGES

The method of steganography outlined in this paper makes use of the more complex regions of an image to embed data. There is no standard definition of image complexity. Kawaguchi discussed this problem in connection with the image thresholding problem, and proposed three types of complexity measures [4][5][6]. In the present paper we adopted a black-and-white border image complexity.

The definition of image complexity

The length of the black-and-white border in a binary image is a good measure for image complexity. If the border is long, the image is complex, otherwise it is simple. The total length of the black-and-white border equals to the summation of the number of color-changes along the rows and columns in an image. For example, a single black pixel surrounded by white background pixels has the boarder length of 4.

We will define the image complexity α by the following.

$$\alpha = \frac{k}{\text{The max. possible B-W changes in the image}} \tag{1}$$

Where, k is the total length of black-and-white border in the image. So, the value ranges over

$$0 \leq \alpha \leq 1. \tag{2}$$

(1) is defined globally, i.e., α is calculated over the whole image area. It gives us the global complexity of a binary image. However, we can also use α for a local image complexity (e.g., an 8×8 pixel-size area). We will use such α as our local complexity measure in this paper.

3. ANALYSIS OF INFORMATIVE AND NOISE-LIKE REGIONS

Informative images are simple, while noise-like images are complex. However, this is only true in cases where such binary images are part of a natural image. In this section we will discuss how many image patterns are informative and how many patterns are noise-like. We will begin by introducing a "conjugation" operation of a binary image.

1. Conjugation of a binary image

Let P be a $2^N \times 2^N$ size black-and-white image with black as the foreground area and white as the background area. W and B denote all-white and all-black patterns, respectively. We introduce two checkerboard patterns Wc and Bc, where Wc has a white pixel at the upper-left position, and Bc is its complement, i.e., the upper-left pixel is black (See Fig. 1). We regard black and white pixels as having a logical value of “1” and “0”, respectively.

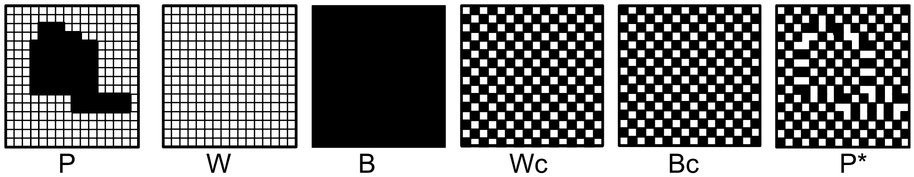


Fig. 1 Illustration of each binary pattern (N=4)

P is interpreted as follows. Pixels in the foreground area have the B pattern, while pixels in the background area have the W pattern. Now we define P* as the conjugate of P which satisfies:

1. The foreground area shape is the same as P.
2. The foreground area has the Bc pattern.
3. The background area has the Wc pattern.

Correspondence between P and P* is one-to-one, onto. The following properties hold true and are easily proved for such conjugation operation. “$\oplus$” designates the exclusive OR operation.

A) $P^* = P \oplus Wc$ (3)

B) $(P^*)^* = P$ (4)

C) $P^* \neq P$ (5)

The most important property about conjugation is the following.

D) Let α (P) be the complexity of a given image P, then we have,

$$\alpha(P^*) = 1 - \alpha(P). \tag{6}$$

It is evident that the combination of each local conjugation (e.g., 8×8 area) makes an overall conjugation (e.g., 512×512 area).

(6) says that every binary image pattern P has its counterpart P*. The complexity value of P* is always symmetrical against P regarding $\alpha = 0.5$. For example, if P has a complexity of 0.7, then P* has a complexity of 0.3.

2. Criterion to segment a bit-plane into informative and noise-like regions

We are interested in how many binary image patterns are informative and how many patterns are noise-like with regard to the complexity measure α.

Firstly, as we think 8×8 is a good size for local area, we want to know the total number of 8×8 binary patterns in relation to α value. This means we must check all 2^{64} different 8×8 patterns. However, 2^{64} is too huge to make an exhaustive check by any means.

Our practical approach is as follows. We first generate as many random 8×8 binary patterns as possible, where each pixel value is set random, but has equal black-and-white probability. Then we make a histogram of all generated patterns in terms of α. This simulates the distribution of 2^{64} binary patterns.

Fig.2 shows the histogram for 4,096,000 8×8 patterns generated by our computer. This histogram shape almost exactly fits the normal distribution function as shown in the figure. We would expect this by application of the central limit theorem. The average value of the complexity α was exactly 0.5. The standard deviation was 0.047 in α. We denote this deviation by σ (“sigma” in Fig. 2)

Secondly, our next task is to determine how much image data we can discard without deteriorating the image quality, or, rather at what complexity does the image data become indispensable.

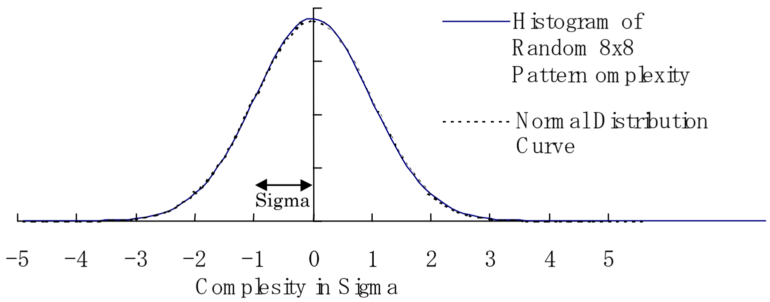


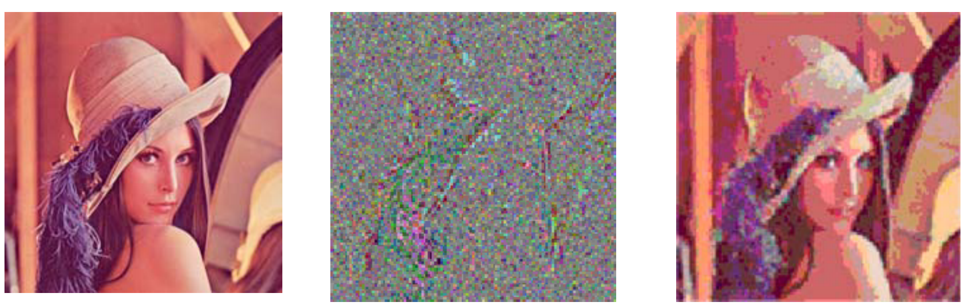
Fig. 2 Histogram of randomly generated 8×8 binary patterns

To discard data means to replace local image areas in a bit-plane with random noise patterns. If we replace all the local areas having complexity value $\alpha_L \leq \alpha$, yet the image still maintains good quality, then perhaps we can discard more. If the quality is no longer good, then we can not discard that much. If $\alpha = \alpha_L$ is the minimum complexity value to be good, such α_L is used as the threshold value.

To be indispensable, or rather "informative," for an image means the following. If the image data is still "picture-like" after we have discarded (randomized) a certain amount of image data for such an α that $\alpha \leq \alpha_U$, and if we discard more, then it becomes only noise-like. Then, that α_U is regarded as the limit of the informative image complexity.

If α_L and α_U coincide ($\alpha_0 = \alpha_L = \alpha_U$), we can conclude α_0 is the complexity threshold to divide informative and noise-like regions in a bit-plane.

We made a "random pattern replacing" experiment on a bit-plane of a color image. Fig. 3 illustrates the result.



A) Original image B) Randomization (simple side) C) Randomization (complex side)

Fig. 3 Randomization of the less and the more complex than $\alpha = 0.5 - 8\sigma$.

Fig. 3 shows that if we randomize regions in each bit-plane which are less complex than $0.5 - 8\sigma$, the image can not be image-like any more. While, we can randomize the more complex regions than $0.5 - 8\sigma$ without losing much of the image information.

This means the most of the informative image information is concentrated in between 0 and $0.5 - 8\sigma$ in complexity scale. Surprising enough, it is only 6.67×10^{-14} % of all 8×8 binary patterns. Amazingly, the rest (i.e., 99.9999999999999333%) are mostly noise-like binary patterns.

The conclusion of this section is as follows. We can categorize the local areas in the bit-planes of a multi-valued image into three portions (1) Natural informative portions (2) Artificial informative portions (3) Noise-like portions.

The reason we categorize the excessively complicated patterns as "informative" is based on our experiments [7].

4. BPCS STEGANOGRAPHY

Bit-Plane Complexity Segmentation Steganography is our new steganographic technique, which has a large information hiding capacity. As was shown in the previous section, the replacement of the complex regions in each bit-plane of a color image with random binary patterns is invisible to the human eye. We can use this property for our information hiding (embedding) strategy. Our practical method is as follows.

In our method we call a carrier image a “vessel” or “dummy” image. It is a color image in BMP file format, which hides (or, embeds) the secret information (files in any format). We segment each secret file to be embedded into a series of blocks having 8 bytes of data each. These blocks are regarded as 8×8 image patterns. We call such blocks the secret blocks. We embed these secret blocks into the vessel image using the following steps.

1. Transform the dummy image from PBC to CGC system.
2. Segment each bit-plane of the dummy image into informative and noise-like regions by using a threshold value (α_0). A typical value is $\alpha_0 = 0.3$.
3. Group the bytes of the secret file into a series of secret blocks.
4. If a block (S) is less complex than the threshold (α_0), then conjugate it to make it a more complex block (S*). The conjugated block must be more complex than α_0 as shown by equation (6).
5. Embed each secret block into the noise-like regions of the bit-planes (or, replace all the noise-like regions with a series of secret blocks). If the block is conjugated, then record this fact in a “conjugation map.”
6. Also embed the conjugation map as was done with the secret blocks.
7. Convert the embedded dummy image from CGC back to PBC.

The Decoding algorithm (i.e., the extracting operation of the secret information from an embedded dummy image) is just the reverse procedure of the embedding steps.

The novelty in BPCS-Steganography is itemized in the following.

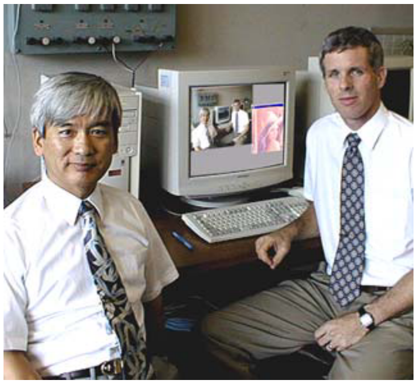
A) Segmentation of each bit-plane of a color image into “Informative” and “Noise-like” regions.
B) Introduction of the B-W boarder based complexity measure (α) for region segmentation
C) Introduction of the conjugation operation to convert simple secret blocks to complex blocks.
D) Using CGC image plane instead of PBC plane

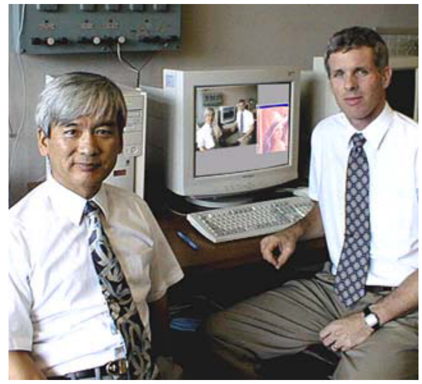
5. EXPERIMENTS

1. Embedding Capacity

We have developed BPCS-Steganography programs for both Windows and Unix. In each program, we took an 8×8 square as the local image size. Fig. 4 (A) is an example of the original dummy image (640x588, full color). (B) is the same image with all the information of Fig. 5 embedded in it. As indicated in Fig. 5 this embedded information includes a picture of Lincoln, the text from four historical documents, and the entire script from seven of Shakespeare’s plays. Note that the size of the embedded information before compression is almost as great as the image size itself. Furthermore, the embedding operation does not increase the size of the image by even a single byte. Yet, even when viewed on the computer monitor, the images before and after embedding are almost indistinguishable from one another.

It should also be noted that the image of Fig. 4 contains a number of large regions that are relatively flat in color. Our BPCS technique made little use of such regions for embedding, as doing so would introduce noticeable noise in these regions. Fig. 6 presents an example of embedding in a scene with few flat regions. In this case the image is 617x504 pixels. (A) shows the original image, and (B) shows the image after embedding all the information of Fig. 5 plus an additional Shakespearean play, “Antony and Cleopatra” of size 179,900 bytes before compression and 64,184 bytes after. Therefore the total information embedded in this 933,408 byte image is actually 1,212,744 bytes before compression; i.e., the embedded information exceeds the vessel size by 30%! The compressed data size is 505,502 bytes, which is 54% of the vessel size. Even with this much information embedded in the image, the embedded and original images look nearly identical when viewed on the monitor.





(A) Original vessel image (B) Embedded vessel image

Fig.4 Example of a vessel image

File	Original Size	Compressed Size
Lincoln Picture at right (jpg)	66,190	66,044
The Gettysburg Address	1,502	742
The Declaration of Independence	9,553	4,075
The Constitution (with amendments)	56,989	14,803
The Magna Carta	31,285	12,089
Romeo and Juliet	149,097	58,829
Hamlet	188,626	74,690
Macbeth	109,281	43,698
A Midsummer Night's Dream	99,623	40,242
The Taming of the Shrew	128,385	49,787
The Tempest	102,788	42,044
A Comedy of Errors	89,525	34,275
Total	**1,032,844**	**441,318**

(A) Summary of embedded data (B) Embedded picture

Fig.5 Files embedded in Fig 4(B)

(A) Original vessel image (B) Embedded vessel image

Fig.6 Example of a vessel image with fewer flat regions

Through our embedding experiments, we found that most color images taken by a digital camera can be used as vessel images. In almost all cases, the information hiding capacity was nearly 50% of the size of each vessel image. This capacity is 4 to 5 times as large as currently known steganographic techniques.

For a given image, embedding capacity can be traded with image quality by altering the complexity threshold. The image of Fig. 4 used a threshold of 24 border pixels per 8×8 region; therefore regions having more border pixels than this were eligible for embedding. Fig. 7 shows how the capacity changes with threshold for this image. For this image a threshold of 24 seemed optimal, while lower thresholds introduced some "noise" to the image.

Threshold	**Capacity**	**Percent Of original**
20	499432	44%
25	469480	41%
30	437240	38%
35	400848	35%
40	361800	32%
45	315432	27%

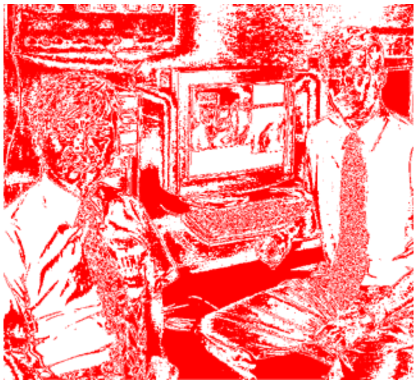
Fig. 7 Capacity vs. Complexity Threshold for the image of Fig. 4

2. Using Gray Coded Bit-Planes for Complexity Segmentation

Fig. 8 illustrates the advantage to using Gray Coded bit planes for complexity segmentation. Parts A through C of this figure show the PBC red bit planes numbered 3 through 5 for the image of Fig. 5a, while parts D through F show the CGC version of these same planes. From looking at such bit planes, one can get a pretty good idea of which regions of the bit plane are complex enough to be replaced with information during BPCS embedding. Recall that the goal with BPCS Steganography is to use as much of the image as possible for hiding information without appreciably altering the visual appearance of the image.

In comparing these two sets of bit planes, it is evident that the PBC bit planes provide a much greater region for embedding. However, substantial portions of the regions on the higher bit planes deemed embeddable using PBC are actually relatively flat in color. For example, note the wall in the background. This is because of the "Hamming Cliffs" which occur with PBC wherein a small change in color affects many bits of the color value. If embedding were to replace the bits in such complex-looking but actually relative flat regions, then substantial color changes would occur. As a simple example, consider a region where the blue value hovers nearly randomly between the binary values of 01111111 and 10000000. In this region, every bit plane would look complex and would thus appear to be embeddable, while in practice, it would be prudent to only embed in the lower one or two planes. Although occurrences such as this where all bits change in a relatively flat region are rare, the frequency of occurrence doubles on each lower bit plane.

CGC images do not suffer from such Hamming Cliffs. Regions which are relatively flat exhibit fewer changes on the higher bit planes. Although this limits the amount of space available for embedding, it does so in regions that should not be altered in the first place. With CGC, embedding in each region is done on the higher bit planes only to the extent allowed by the complexity produced by actual color variation.

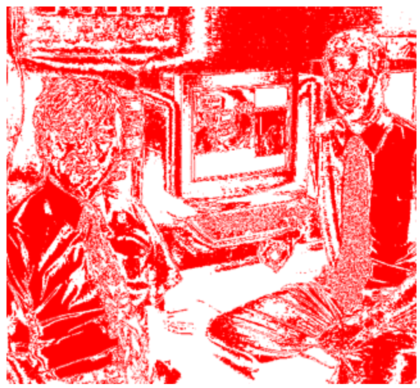


(A) PBC red plane 3

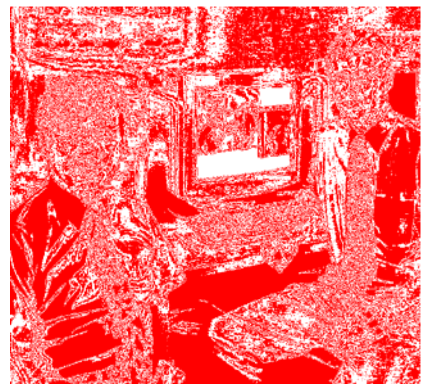
(B) PBC red plane 4

(C) PBC red plane 5

(D) CGC red plane 3



(E) CGC red plane 4



(F) CGC red plane 5

Fig. 8 Comparison of PBC and CGC bit planes

6. CUSTOMIZATION OF THE PROGRAM

The BPCS-Steganography algorithm has several embedding parameters for a practical program implementation. Some of them are:

(1) The embedding location of the header(s) of the secret file(s)
(2) The embedding threshold, α_0.
(3) The sequence in which the 8×8 regions of the vessel image are considered for embedding.
(4) The encoding of the conjugation map.
(5) Special operations, such as an exclusive-or of the header bytes or embedded data with pseudo-random numbers.
(6) Encryption parameters of the secret file(s)
(7) The compression parameters of the secret file(s)

It is very easy for a single BPCS Steganography program to allow the user to customize parameters such as these, producing a very large number of possible customized programs. In this way, each user or group of users can have their own program that embeds data in an image in a way that is unreadable by others.

7. APPLICATIONS

In discussing applications of BPCS Steganography, it is instructive to note that it differs from digital watermarking in two fundamental ways. The first is that for full color (e.g., 24-bit) images, it has a very large embedding capacity. As described previously, our experiments with BMP images have shown capacities exceeding 50% of the original image size. Although the results presented in this paper are for 24-bit images, we have also been working with other formats, such as 256 color images, which utilize a palette. Although the capacity is lower, the same concepts can be applied.

The second difference is that BPCS Steganography is not robust to even small changes in the image. This can be viewed as a good thing in applications where an unknowing user might acquire an embedded image. Any alteration, such as clipping, sharpening or lossy compression, would "destroy the evidence" and make it unusable for later extraction. Extracting the embedded information requires a deliberate attempt by a knowledgeable user on an unaltered image. The lack of robustness also ties in to the fact that a malicious user cannot alter the embedded data without knowledge of the customization parameters.

The more obvious applications of BPCS Steganography relate to secret communications. For example a person, group, or company can have a web page containing secret information meant for another. Anyone can download the web page, so when the intended recipient does so, it does not draw any attention. Extracting the embedded information would require software customized with the proper parameters. Encryption of the embedded data would further improve security. This scenario is analogous to putting something in a very secure safe and then hiding the safe in a hard to find place.

In some applications, the presence of the embedded data may be known, but without the customization parameters, the data is inseparable from the image. In such cases, the image can be viewable by regular means, but the data is tied to the image and can't readily be replaced with other data. Others may know the data is there, but without the customization parameters, they cannot alter it and still make it readable by the customized software.

Applications of BPCS Steganography are not limited to those related to secrecy. For such applications, the presence of the embedded data may be known, and the software for extraction and embedding can be standardized to a common set of customization parameters. An example of this is a digital photo album, where information related to a photo, such as date and time taken, exposure parameters, and scene content, can be embedded in the photo itself.

8. CONCLUSIONS AND FUTURE WORK

The objective of this paper was to demonstrate our BPCS-Steganography, which is based on a property of the human visual system. The most important point for this technique is that humans can not see any information in the bit-planes of a color image if it is very complex. We have discussed the following points and showed our experiments.

(1) We can categorize the bit-planes of a natural image as informative areas and noise-like areas by the complexity thresholding.
(2) Humans see informative information only in a very simple binary pattern.
(3) We can replace complex regions with secret information in the bit-planes of a natural image without changing the image quality. This leads to our BPCS-Steganography.
(4) Gray coding provides a better means of identifying which regions of the higher bit planes can be embedded.
(5) A BPCS-Steganography program can be customized for each user. Thus it guarantees secret Internet communication.

We are very convinced that this steganography is a very strong information security technique, especially when combined with encrypted embedded data. Furthermore, it can be applied to areas other than secret communication. Future research will include the application to vessels other than 24-bit images, identifying and formalizing the customization parameters, and developing new applications.

9. ACKNOWLEDGEMENT

This project is partly funded by the Advanced Information Technology Program (AITP) of Information-technology Promotion Agency (IPA), Japan.

10. REFERENCES

1. Hall, Ernest L., *Computer Image Processing and Recognition*, Academic Press, New York, 1979.
2. Jain, Anil K., *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.
3. Kawaguchi, E., Endo, T. and Matsunaga, J., “Depth-first picture expression viewed from digital picture processing”, *IEEE Trans. on PAMI*, vol.5, no.4, pp.373-384, 1988.
4. Kawaguchi, E. and Taniguchi, R., “Complexity of binary pictures and image thresholding - An application of DF-Expression to the thresholding problem”, *Proceedings of 8th ICPR*, vol.2, pp.1221-1225, 1986.
5. Kawaguchi, E. and Taniguchi, R., “The DF-Expression as an image thresholding strategy”, *IEEE Trans. on SMC*, vol.19, no.5, pp.1321-1328, 1989.
6. Kamata, S, Eason, R. O., and Kawaguchi, E., “Depth-First Coding for multi-valued pictures using bit-plane decomposition”, *IEEE Trans. on Comm*., vo.43, no.5, pp.1961-1969, 1995.
7. Kawaguchi, E. and Niimi M, “Modeling Digital Image into Informative and Noise-Like Regions by Complexity Measure”, Preprint of the *7th European-Japanese Conference on Information Modeling and Knowledge Bases*, pp.268-278, May, Toulouse, 1997.