Embedding an Image Into Image by Region Segmentation Method Based on a Complexity With Effect of Standard Deviation

Manoj Chavan
Department of EXTC
Thakur College of Engineering and Technology
Mumbai, India

Nitin Sambre
Department of EXTC
KIT College of Engineering
Kolhapur, India

Aradhana Manekar
Department of EXTC
Thakur College of Engineering and Technology
Mumbai, India

ABSTRACT
In this paper, an image steganography system implemented is described, in which the data hiding (embedding) is achieved using image segmentation based on a local complexity measure. A complexity measure is defined in Steganography for discriminating noisy regions in an image. This paper presents a revised method for embedding data into an image by variation in complexity threshold measure and effect of standard deviation of various bit planes. The principle of the method is based on that of bit plane complexity based steganography. High embedding rates are achieved with low distortion based on the theory that noise-like regions in an image's bit-planes can be replaced with noise-like secret data without significant loss in image quality Percentage embeddable area of complex images is around 50% without visual distortion showing PSNR above 30dB. While computing complexity threshold minimum standard deviation of the plane should be selected

Categories and Subject Descriptors
I.4.m Miscellaneous
Image processing and computer vision

General Terms
Design, Measurement, Performance, Verification.

Keywords
Steganography ,image embedding, CGC

1. INTRODUCTION
In this paper, a technique to embed secret data into a dummy image by using image segmentation based on a local complexity measure is proposed and effect of standard deviation of various bit planes is shown. The key idea to this approach is that a binary image can be categorized as "informative" and "noise-like" regions. Each region is segmented by some "complexity

measure." Human visual perception is not affected by a replacement of noise like data with other data which also looks noise like. This means that if the embedding data is noiselike, then it can be hidden in the noiselike regions of a digital image. If a part of embedding data is simple, i.e., looks informative, then apply "image-conjugate" operation to it. This operation transforms a simple pattern into a complex pattern. Then encrypt the secret data. The encrypted data can be recovered by separating the dummy part through a complexity segmentation method.[1][2]

2. Gray Coded Bit-Planes for Complexity Segmentation
PBC (Pure-Binary Coding) for the image is shown in Figure 1(a), while parts 1(b) show the CGC (Canonical Gray Coding) 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 embedding The goal with this 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 in Figure1, 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 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 red 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. Regions which are relatively flat exhibit fewer changes on the higher bit planes.

Figure 1. (a) bit plane 2 of PBC (Pure-Binary Coding)
(b) bit plane 2 of CGC (Canonical Gray Coding)
(c)PBC (Pure-Binary Coding)
(d) CGC (Canonical Gray Coding)

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.

3. BLOCK COMPLEXITY MEASURE.

The complexity of a binary image can be defined case by case, because there is no standard definition of image complexity. In this paper, complexity measure defined by the length of black and white border is used[3].

3.1 Calculation of Complexity measure α for binary images

The four-connectivity neighborhood method is used. In that case, the total length of black-and- white border equals to the summation of the number of color-changes along the rows and columns in the image. For example, a single black pixel surrounded by white background pixels has the boarder length of

4. We assume the image frame is always square having 2" x 2m pixels (Practical image size is m = 8 N 12). The number of the color-changes in the interior area of the image is counted. Therefore, the minimum of the border length is 0 (either black or white pattern), while the maximum is 2 x 2^m x (2^m - 1) Thus, the image complexity measure is defined by the following.

$$\alpha = \frac{K}{2 \times 2^m \times (2^m - 1)} \quad (1)$$

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

$$0 \le \alpha \le 1 \quad (2)$$

Equation (1) define α for a local image complexity (e.g., an 8 × 8 pixel-size for m=3), such α are used as local complexity measure.

3.2 Dynamic allocation of complexity threshold measure.

In order to check statistical properties of 8 x 8 regions, histogram of α of all the bit planes was plotted. This histogram shape almost exactly fit the normal dist α distribution curve. The average value of the complexity in this histogram was exactly 0.5. This is deviation is denoted by σ. Then, the critical value of α for informative and noise like image segmentation is examined by replacing all the $2^3 \times 2^3$ areas having complexity value α0 with random noise patterns .The smallest α is regarded as the complexity threshold. for informative/noise-like segmentation. A noise replacement experiment have been made using 8 bit/pixel gray scale images. This image is decomposed into 8-bitplanes by bit-slicing (by Canonical Gray Coded system). Setting the threshold at α= 0.5-4σ. Experiments were carried out and these experiments showed that the "informative "and "noise-like" criterion for $2^3 \times 2^3$ size area is around α = 0.5 - 4σ for 8 bit images. The value of σ is calculated for all the bit planes and the smallest value of σ is dynamically allotted to the α0 for threshold complexity measure.

The most important result from this experiment was as follows. If the secret data, which we want to encrypt, can be treated as a random (i.e., noise-like) binary image, and if the α of each local area satisfies0.5 - 4σ ≤ α, the secret image is embedded into these areas of a dummy image.

3.4 Conjugation of non noisy secret data.

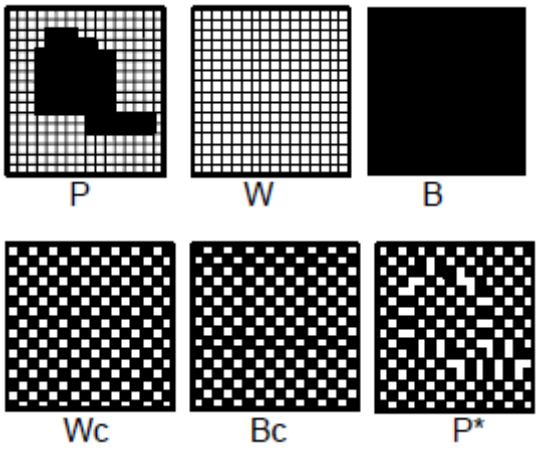

Figure 2 Illustration of each binary pattern [3]

After thresholding of secret image if the segmented block is not found complex then it is made complex by conjugation operation as follows[3].

Let P be a 2m x 2m size black-and-white image with black area **as** the foreground and white area **as** the background area. W and B denote all white and all black patterns, respectively. Two checkerboard patterns Wc and Bc are introduced, where W, has a white pixel at the most upper-left position, and B, is its complement, i.e., the most upper-left pixel is black. P can be interpreted as pixels in the foreground area which have B pattern, while pixels in the background area have W pattern.

Now P* defined 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. "⊕" designates the exclusive OR operation.

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

$$(P^*)^* = P \quad (4)$$

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

The most important property about conjugation is the following, Let α (P) be the complexity of a given image P, then we have,

$$\alpha(P^*) = 1 - \alpha(P). \quad (6)$$

Equation (6) says that every binary image pattern P has its counterpart P*. The complexity value of P* is always symmetrical against P regarding α = 0.5. For example, if P has a complexity of 0.7, then P* has a complexity of 0.3. It is evident that the combination of each local conjugation (e.g., 8 × 8 area) makes an overall conjugation (e.g., 512 × 512 area). This is useful for data embedding, because the original image can be locally conjugated to get a modified α. In that case, however, local "conjugation-map" must be kept, which describe the location of image conjugation to recover the original image.

3.3 Embedding and retrieval method using Complexity Segmentation

Embed each secret block into the noise-like regions of the bit-planes by doing complexity thresholding (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." Also embed the conjugation map as was done with the secret blocks. Convert the embedded dummy image from CGC back to PBC. For color images separate R-G-B components and treat each component as a separate image

The extraction of the secret data from embedded image is performed by the reverse steps. It is impossible to extract it without knowing complexity threshold and conjugation-map.

4. Evaluation criteria used for comparison of parameters

4.1 Distortion measure.

Generally the image distortion is measured numerically using PSNR [3] (Peak Signal Noise Ratio), which is given by

$$PSNR = 10 log_{10}\left(\frac{256^2}{MSE}\right)$$

$$MSE = \frac{\sum_{N2}^{N1}(Ic - Is)}{N1N2} \quad (7)$$

Ic , Is are the cover image and the stego image, respectively.

4.2 Capacity Measure.

4.2.1 Bts per pixel

The notion of capacity in data hiding indicates the maximum number of bits that can be hidden and successfully recovered by the steganographic system. Because of that the number of hidden bits varies depending on cover image size, to measure the hidden capacity, we use bits per- pixel bpp given by

$$bpp = \frac{hidden\ bits}{Numpix(I_c)} \quad (8)$$

where Numpix(Ic) is total pixels number of pixels in the cover image.

4.2.2 Embedding Capacity

This can be found by calculating the number of complex blocks of embeddable area in the vessel image after complexity thresholding [3].

$$\mathbf{E_c = Total\ Complex\ Blocks\ x\ 2^m\ x\ 2^m\ bits} \quad (9)$$

Where m = 2 or 3 depending upon the block size

4.2.3 Embeddable area.

Percentage embeddable area in a vessel image can be calculated by taking ratio of Embeddable area size to total vessel image size

5. Result.

Shown below are some snapshots of results taken in Matlab (Ra2007 version).

For statistical analysis purpose selected test images are shown in Figure.8. Baboon, Lena and Forest are used as vessel image whereas Ganpati image is used as secret image to be embedded. Baboon and Forest images are complex in nature, while Lena show comparatively less complexity and are thus simple in nature. Thus the effect on various parameters could be shown properly with the selected images. Thus they are so chosen.

Figure 3. Test Images: a) Baboon Image b) Lena Image c) Forest Image d) Ganpati Image

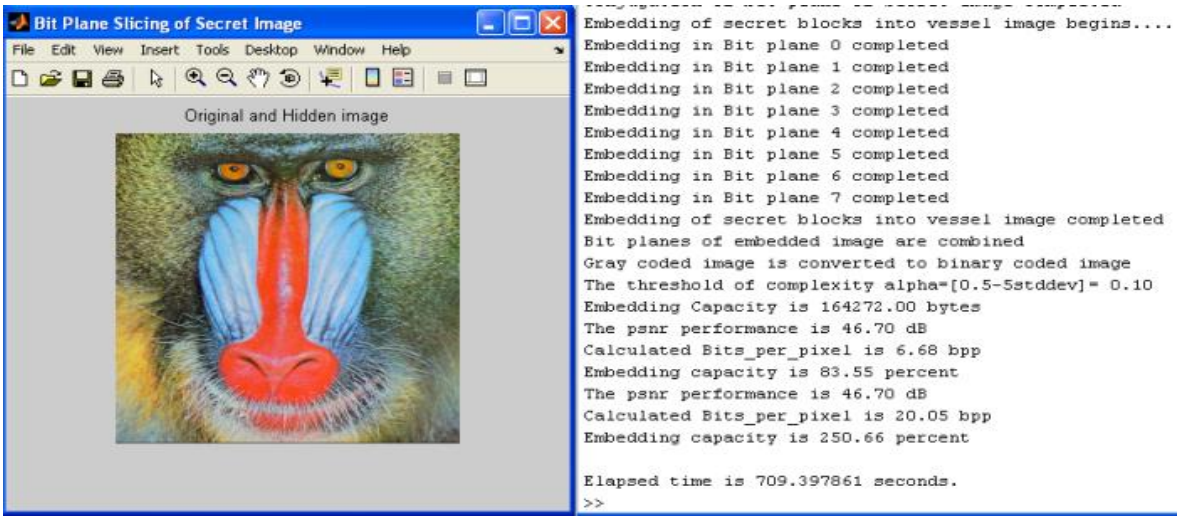

Figure 4. 128x128 size secret image Embedded into 256x256 vessel image

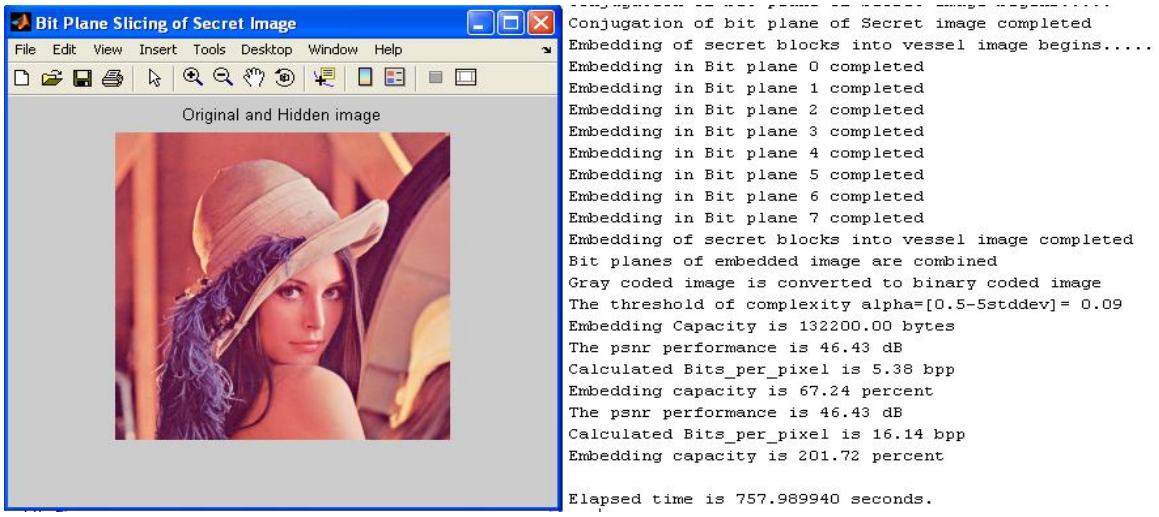

Figure 5. 128x128 size secret image Embedded into 256x256 vessel image

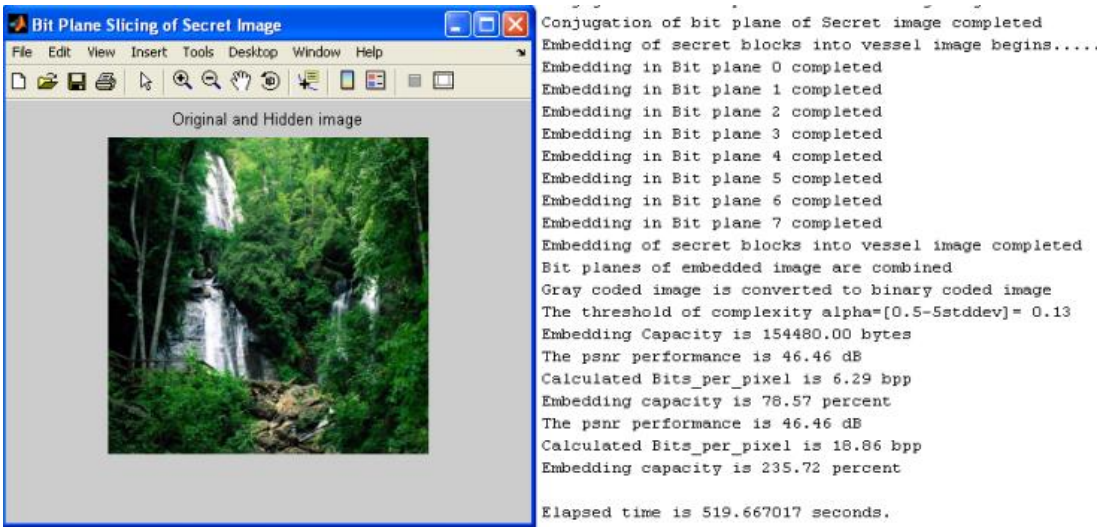

Figure 6. 128x128 size secret image Embedded into 256x256 vessel image

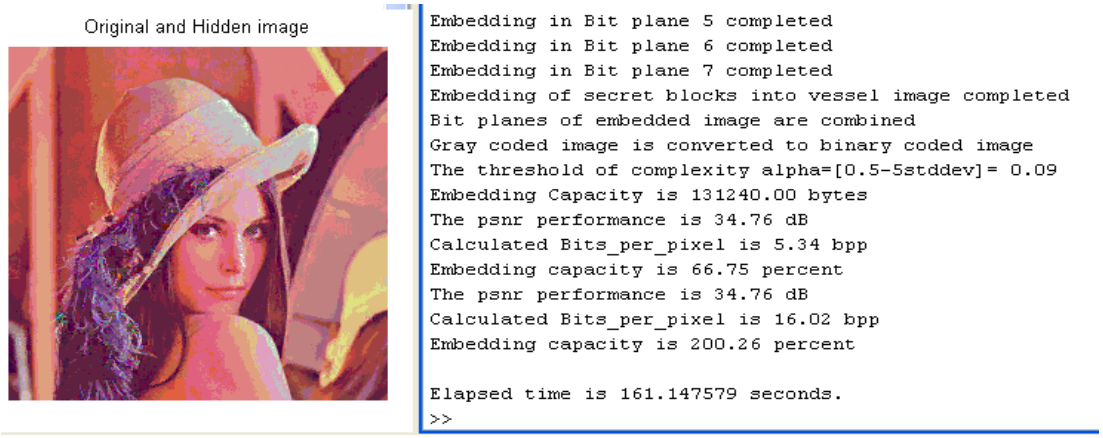

Figure 7. 192x192 size secret image Embedded into 256x256 vessel image

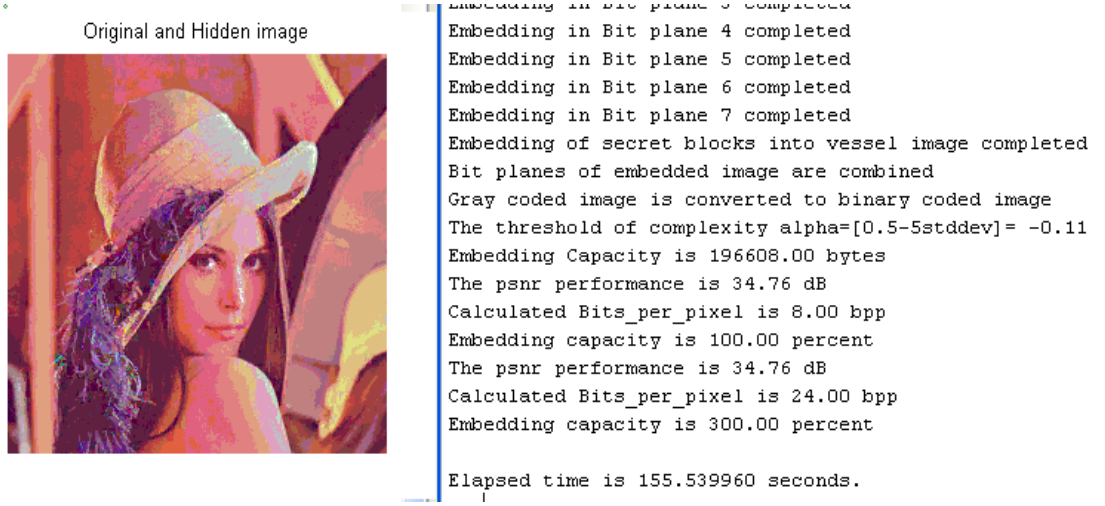

Figure 8. 192x192 size secret image Embedded into 256x256 vessel image

Table 1 Result Table For Comparison of effect of standard deviation with Vessel Image of 256 X 256 and Secret Image 128 X 128

Vessel Image Size : Baboon (256X256)					
Secret image size : Ganpati (128X128)					
α	**Embedding Capacity in bytes**	**PSNR dB**	**Embeddable Bpp**	**% Embeddable Area**	**Threshold**
α_{min}	127232	46.72	15.53	194.14	0.31
α_{avg}	164272	46.70	20.05	250.66	0.10
α_{max}	196608	46.69	24.00	300.00	-0.08

Table 2 Result Table For Comparison of effect of standard deviation with Vessel Image of 256 X 256 and Secret Image 128 X 128

Vessel Image Size : Lena (256X256)					
Secret image size : Ganpati (128X128)					
α	**Embedding Capacity in bytes**	**PSNR dB**	**Embeddable Bpp**	**% Embeddable Area**	**Threshold**
α_{min}	86536	46.48	10.56	132.04	0.30
α_{avg}	132200	46.43	16.14	201.72	0.09
α_{max}	196608	46.44	24.00	300.00	-0.11

Table 3 Result Table For Comparison of effect of standard deviation with Vessel Image of 256 X 256 and Secret Image 128 X 128

Vessel Image Size : Forest (256X256)					
Secret image size : Ganpati (128X128)					
α	**Embedding Capacity in bytes**	**PSNR dB**	**Embeddable Bpp**	**% Embeddable Area**	**Threshold**
α_{min}	128752	46.5	15.72	196.46	0.28
α_{avg}	154450	46.46	18.86	235.72	0.13
α_{max}	188960	46.45	23.07	288.33	-0.02

6. CONCLUSION.

As the complexity of image increases (found by thresholding), the embeddable area also increases. Baboon and forest images show higher complexity and thus higher embeddable capacity than Lena. The embedding capacity of color image is almost triple the embedding capacity of gray scale images .PSNR calculated in the result table is calculated after embedding fixed size of image and not full embedding capacity. Generally PSNR above 30 is considered to be good and Table1, 2, 3 show very good PSNR after embedding secret images of different sizes. Embeddable Bits per pixel for Baboon and Forest are almost fifteen i.e. out of twenty four bits fifteen bits can be used for embedding. Embeddable Bits per pixel for Lena are almost ten i.e. out of twenty four bits ten bits can be used for embedding. There is no change in the size of the vessel image after embedding. Selecting minimum standard deviation of the bit plane of the vessel image for deciding complexity threshold is most proffered, average standard deviation will result into increase in embedding capacity but will not be imperceptible whereas maximum standard deviation will result into threshold being almost zero there by replacing all pixel of vessel image thereby making it improper for use.

7. REFERENCES

[1] N.F.Johnson and S.Jajodia, "Exploring Steganography: Seeing the Unseen," IEEE Computer, 2 1998, pp.26–34.

[2] C. Cachin, "An information theoretic model for steganography", Lecture notes in Computer Science, vol. 1525, 1998, pp. 306-318, Springer-Verlag..

[3] E.Kawaguchi and R.O.Eason, "Principle and applications of BPCS-Steganography," P rocee dings of SPIE: Multimedia Systems and Applications, vol.3528, 1998, pp.464–472

[4] Kawaguchi, E., Endo, T. and Matsunaga, J., "Depth First picture expression viewed from dig- ital picture processing", IEEE Trans. on PAMI, vol.PAM1-5, n0.4~pp.373-384, 1988.

[5] Peticolas F A P, Anderson R J, Kuhn M G. Information Hiding-A Survey [J].Proceeding of the IEEE, 1999, 87(7):1062~1078.

[6] Irfanview image analiser,version 4.27, 2010 http://www.irfanview.com/

[7] Silvia Torres-Maya, Mariko Nakano-Miyatake and Héctor Perez-Meana" An Image Steganography Systems Based on BPCS and IWT" Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP 2006) 0-7695-2505-9/06 © 2006 IE

[8] R.C.Gonzalez, R.E.Woods, "Digital Image Processing using MATLAB",Pearson Education, 2004