NEW PIXEL SORTING METHOD FOR PALETTE BASED STEGANOGRAPHY AND COLOR MODEL SELECTION

Sos S. Agaian[1] and Juan P. Perez[2]

[1]NSP Lab, The University of Texas at San Antonio, Electrical Engineering Department,
6900 North Loop 1604 West, San Antonio Texas 78249, sagaian@utsa.edu
[2] NSP Lab, The University of Texas at San Antonio, Electrical Engineering Department,
6900 North Loop 1604 West, San Antonio Texas 78249, jpperez@lonestar.utsa.edu

ABSTRACT

In this article, we propose a new steganographic approach for palette-based images. This new method has the advantage of embedding secure data, within the index, the palette or both, using special sorting scheme. In the index, information can be embedded using a double pseudo random key that selects blocks and pixels within the image randomly. A stego-bit is embedded at each pixel of the selected blocks. Embedding is accomplished by selecting the closest color using a weighted distance measure that contains the desired modulus-bit. This algorithm also gives you the option of selecting the best color models (i.e. YCbCr and HSV) for embedding purposes. The presented technique also incorporates the use of a cover image measure in order to select the best of the candidates for the insertion of the stego information. This has the advantage of introducing lower distortion and higher embedding capacity than EzStego and Fridrich's method. Computer simulations have shown that the presented algorithms carry with them the following advantages: 1) Introduces 8-10 times less distortion than EzStego [4] and 4-5 times less distortion than Fridrich's [3] method. 2) Provides additional security through a simple selective color and cover image algorithm. 3) Provides the closest color substitutes using different color models and a weighted distance measure. 4) Finally, it offers an increased capacity by embedding only on the best cover images.

1. INTRODUCTION

Steganography is the technique used to hide the presence of information in inconspicuous looking carriers. In order make the communication more secure, the secret information can be encrypted before it is hidden in the carrier. Encrypting the message before hiding is suggested and gives double security. There are different algorithms for hiding information that include the usage of audio, video, and images with different image formats, i.e. JPEG, GIF, BMP and TIF.

In today's world more information is transmitted and stored using computers and the internet. The usage of such systems as means of communication has become a necessity rather than a commodity. As the need for electronic data increases so does the need to protect it. One way of securing data is by hiding it in inconspicuous looking carriers. There are many forms that take advantage of the gaps in the human visual and audio systems to hide information. Digital images contain considerable amounts of unnecessary data to the human visual system; therefore they are ideal carriers used to conceal the presence of hidden information. In order to embed information, it is necessary to have a cover image (image used for embedding) and a stego message (secret information).

There are several algorithms that are used to embed in palette based images. Some examples of the well known algorithms are: Gifshufle [7], EzStego [4], Fridrich's parity bit method [3], BPCS Steganographic method for palette-based images [5,6] and the new pixel sorting method [1]. These algorithms have several limitations which are explained in section 2.2.

This paper is structured in using the following format: Section 2 discusses the background on color modeling. Section 3 defines palette-based steganography and describes existing methods and problems introduced by some of these methods. The new method is introduced in Section 4 as a solution to some of the existing problems. This section also provides the details involving: a better pixel sorting method; an improved color selecting scheme; and a new method for selecting the block size that introduces less distortion.

2. BACKGROUND

The purpose of color models is to organize colors in a standard form. Different models are used according to the user's need. These models are divided in two models: hardware oriented and color manipulation. The color models include RGB, CMY, CMYK, HSI, HSV, RGB and YIQ.

In this paper we will focus on the YCbCr and HSV color models. Other models can also be used however we found that the two mentioned models work the best. In the first model the Y stands for luminance, Cb for chrominance blue and Cr for chrominance red. In the second model the H stands for hue, the S for saturation and the V for value.

In this article the YCbCr and HSV color models are used to sort color palettes. In fact these models make it easy to separate the image's information from the color information. This way for example when we look for a color using the distance formula, we would put more emphasis on the luminance if we were using the YCbCr model. On the other hand if we were using the HSV we would emphasize the value (V) in order to select the best color. In figure 1 we can see the differences between the different color models

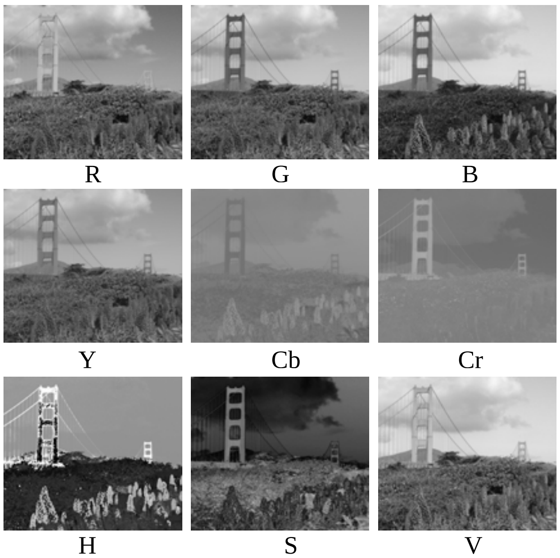


Figure 1. The image GoldenGate is represented using the RGB, YCbCr and HSV color model.

$$V = \max(R, G, B) \qquad (1)$$

$$S = \begin{cases} 0 & \text{if } \max = 0 \\ \dfrac{\max - \min}{\max} & \text{otherwise} \end{cases} \qquad (2)$$

Formula for H

$$\begin{cases} \dfrac{(g-b)}{(\max - \min) \times 60} & \text{if } (r = \max \wedge (g-b) > 0) \\ \dfrac{(g-b)}{(\max - \min) \times 60 + 300} & \text{if } (r = \max \wedge (g-b) < 0) \\ 2 + \dfrac{(b-r)}{(\max - \min)} \times 60 & \text{if } (g = \max) \\ 4 + \dfrac{(r-g)}{(\max - \min)} \times 60 & \text{if } (b = \max) \end{cases} \qquad (3)$$

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.419 & -0.081 \\ -0.169 & -0.331 & 0.500 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad (4)$$

3. PALETTE-BASED STEGANOGRAPHY

This section describes basic methodology for embedding using palette-based images. In addition, we also introduce other exiting methods. Finally, some the most common problems with these algorithms are discussed.

First of all, why do we want to use palette-based images? In order to answer this question we have to understand what palette-based images are. These types of images have a resolution of 8-bits/pixel. These images can be transformed from a three color layer image by reducing the number of unique colors used within an image. The colors are reduced by using color quantization. The main idea of this process is to take multiple colors with minimum variations and group them to make one new color. This process is repeated until the number of unique colors is 256 or less. There are many procedures for color quantization. After the quantization the colors are mapped to the index.

In order to know how the information can be hidden in these images it is essential to understand how these formats are structured. These images consist of an index and a palette. The index, for example, contains information specifying where each color is located in the palette. The in index is illustrated in figure 2, in this figure we took a small block and expanded it in order to show the pixels.

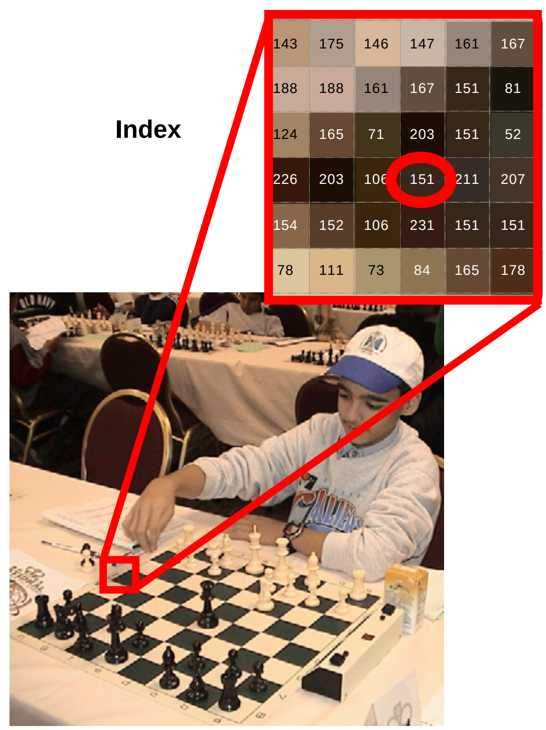


Figure 2. This figure shows the indices of the image. This indices point to colors in the palette (shown in figure 3).

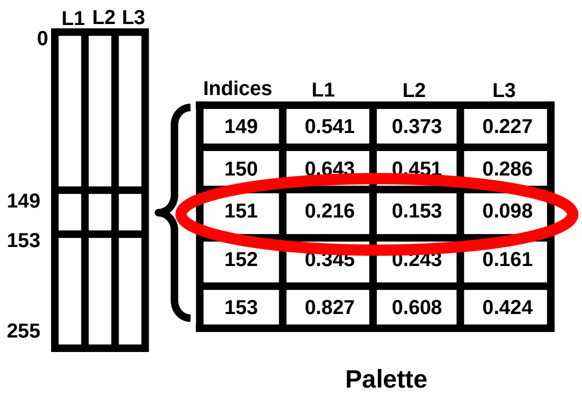


Figure 3. This figure shows the palette of the image. The palette is an array that contains all of the colors used in the image.

The palette on the other hand, is an array consisting of three columns and N rows, where N is the number of unique colors. It also contains all the colors used in the image; each of the three columns in the palette corresponds to the color components. In figure 3, we denote the color layers using the symbols L1, L2 and L3. The color layers depend on the color representation used. Each of these colors is a combination of normalized RGB values ranging from 0 to 1.

Palette-based images are used as cover images to provide a secure and fast transmission/storage over a communication system. Palette-based images are largely available on the Internet in the GIF format. Due to their abundance over the Internet it is difficult to find a suspicious stego-image. Furthermore, since there was some alteration introduced by the color quantization, the stego message is able to pass as noise. Given that palette-based images have a smaller resolution, they can be transmitted faster than 24-bit resolution images. In these algorithms, information has been embedded in two ways: manipulating either the palette or the image index.

The basic idea of embedding in the palette lies in the insertion of secret messages within the ordering of the colors in the color-map. This avoids changes in the image leaving the visual perception of the image unscathed. This is possible because two identical images may have completely different color-maps. It is natural that the information can also be embedded using pre-selected pixels and arranging the palette in a structure where neighboring colors are close in given distance, including Euclidian distance. It is better if the secret message is embedded in the image's index, since the palette provides limited capacity and questionable security.

3.1 Existing Steganographic Methods

There are several methods that embed using these formats. Some of the most popular existing methods for palette-based steganography include: Gifshufle [7], EzStego [4], Fridrich's parity bit method [3], BPCS Steganographic method for palette-based images [5,6] and the new pixel sorting method [1].

Gifshuffle is an algorithm that reorders the color palettes in order to hide information. The following formula is used to sort the palettes and remember the order.

$$X = R * 65536 + G * 256 + B \qquad (5)$$

According to [7], a palette with 256 rows can store up to log2 (n!) or in this case 1675 bits. The extraction is done the same way: first the palette is sorted shuffled (stego-palette).

The palette-based methods developed by Fridrich [3] and Machado's EzStego [4] embed in the image's index. EzStego [4], for example, is a non-adaptive method that embeds in the LSB of the index. After embedding the image is reconstructed by arranging the palette.

Fridrich's palette-steganographic algorithm is also a non adaptive method, but instead this scheme changes the image's index. This algorithm organizes the pixels in the index of the image instead of changing the least significant bit. This algorithm uses two main equations the distance formula and modulus formula. The modulus formula is used to embed since all the colors in the palette contain either a parity bit of 1 or 0. The distance formula allows the algorithm to constantly look for the colors that are closest in the color space.

Other existing methods embed like [5,6] use color quantization. This method uses a bit plane complexity measure in order calculate the degree of significant information contained within binary image pattern.
The identified noise-like regions are then replaced with the stego-message. In addition, this method utilizes a color quantization to prevent the image from exceeding 256 colors.

3.2 Existing Methods' Flaws

Problems with existing methods that embed within the palette are that they do not take in to account other important color models. Also, the information is limited and the hidden message can be destroyed by switching the order of the palettes. A solution to this would be to use the index to hide the information. This, however, will introduce distortion unless the palette is arranged. The palette has to be arranged in such a way that if the selected pixel changes, the new pixel will point to a similar color. Romana Machado solved this problem by arranging the palette according to its luminance values. This introduces another difficulty; luminance values are calculated using a linear combination of the R, G and B values. The problem is that if the same values appear in a different combination the luminance will remain even if the colors are different.

Fridrich introduced a solution to the problem with a new algorithm that hides message bits into the parity bit of close colors. In this algorithm, Fridrich uses distance to select pixels that are close in distance. This algorithm has the following disadvantages: 1) The embedding capacity is limited to the size of the index. 2) This algorithm is limited to calculating distance using a combination of the three colors R, G, and B versus calculating individual distances. 3) It uses the entire image to find the desired parity bit, providing more room for errors. In order to avoid these problems it is important to introduce a method that calculates the distances between colors using individual values for R, G and B. This is done by selecting special weighted values. Instead of embedding on the whole image this algorithm embeds on different block sizes. In addition, with the used of a simple steganographic capacity measure, the best block size and the best weight values can be selected. By using this technique the best cover image can be selected. This technique introduces less distortion compared to EzStego and Fridrich's parity bit method.

4. NEW PIXEL SORTING METHOD

This selection introduces a new pixel sorting and steganographic capacity measure. This section covers the improved color selection. This section also includes the fact that the steganographic performance can be improved by using different block sizes. In the final subdivision of this section the best color distances and the best block sizes are combined for optimal results.

The primary purpose of this new method is to find the cover image that introduces less noise. The best cover image is found by using a simple capacity measure formula.

$$M = MSE \times \frac{\text{cover image}}{\text{stego message}} \quad (6)$$

This capacity measure is calculated by multiplying the mean square error times the size ratio of the cover image and the stego message. First, the covered image is divided into block sizes. Using the stego-capacity measure, only the best block sizes are selected. Then the message is converted into 8-bit layers. Pixels are selected using a random generated key. Each selected pixel is embedded by replacing it with another that contains the desired stego-bit. Only the pixels which contain the nearest colors are considered. This algorithm searches for the nearest colors by alternating the α, β *and* δ values. The following equations are used to find the stego-bit and nearest color using the best combinations. This method follows the next steps:

Input: Stego-message and set of cover images

Step 1. Divides the image into M x N blocks and chooses the best size

Step 2. Randomly selects pixels from the index

Step 3. Select the best color model

Step 3. Used the Distance formula to select the closest color that contains the desired stegobit and replace the color

$$D = \sqrt{\alpha_2 \left(R_1 - R_2\right)^2 + \beta_2 \left(G_1 - G_2\right)^2 + \delta_2 \left(B_1 - B_2\right)^2} \quad (7)$$

$$Stegobit = (\text{mod}(\alpha_1 \times R + \beta_1 \times G + \delta_1 \times B), 2) \quad (8)$$

Step 4. Select the best cover image using the capacity measure.

Output: Stego image using the best cover image

4.1 Improved Color Selection

In order to enhance the performance of color selection, we used three color models (RGB, YCbCr and HSV) and select the best one for a given image. Then we applied the weighted values (alpha, beta and gamma) to the distance formula (7) and the stego-bit formula (8). Note that these weighted values are independent from each other. Selecting the nearest color in an image using the distance formula can sometimes be vague. This is caused in large part due to fact that the palette has limited colors. In order to improve the quality, color selection has to be done using a distance formula that adapts to the colors of the image. Obviously, if the distance formula is applied to an image that contains large amounts of red, it is going to benefit more by applying more weight to the red value than to the rest of the colors. For this section we ran an experiment alternating the color models and weighted values using different images. The following values were recorded on tables 1 and 2 using the Fridrich's method [2], the best withed values for using the RGB, YCbCr and HSV color models. From this table we can see that the presented algorithms perform better in all models.

Table 1. This table contains the RMS values obtained by comparing the clean and Stego "Fisherman" images

Data	Fridrich	W(RGB)	W(YCbCr)	W(HSV)
5k	0.0041	0.0019	0.0038	0.0036
7k	0.0047	0.0021	0.0043	0.0040
8k	0.0051	0.0023	0.0047	0.0045
13k	0.0064	0.0030	0.0060	0.0056
24k	0.0093	0.0043	0.0085	0.0082
32k	0.0104	0.0048	0.0096	0.0093

Table 2. This table contains the RMS values obtained by comparing the clean and Stego "Sarkis" images

Data	Fridrich	W(RGB)	W(YCbCr)	W(HSV)
5k	0.0038	0.0015	0.0033	0.0034
7k	0.0043	0.0017	0.0037	0.0038
8k	0.0047	0.0018	0.0040	0.0041
13k	0.0059	0.0023	0.0053	0.0053
24k	0.0086	0.0034	0.0074	0.0075
32k	0.0096	0.0038	0.0083	0.0085

4.2 Block Size Selection

In order to further improve the quality of this method's performance, information can be embedded in blocks. The for enhancement purposes the best block is selected. The advantage of embedding by blocks is that the colors in the palette are closer in the color-space and therefore it introduces less distortion.

In this experiment we first embed different text messages, ranging from 2k-32k, in 512x512 images. The embedding was done using different block sizes. The block sizes ranged from 4x4 to 512x512. The best results were produced when embedding in 32x32 blocks. The worst results were produced by embedding in 4x4 or the 512x512. The 512x512 is equivalent to using Fridrich's [2] method. In this case our method proved to be better than Fridrich's.

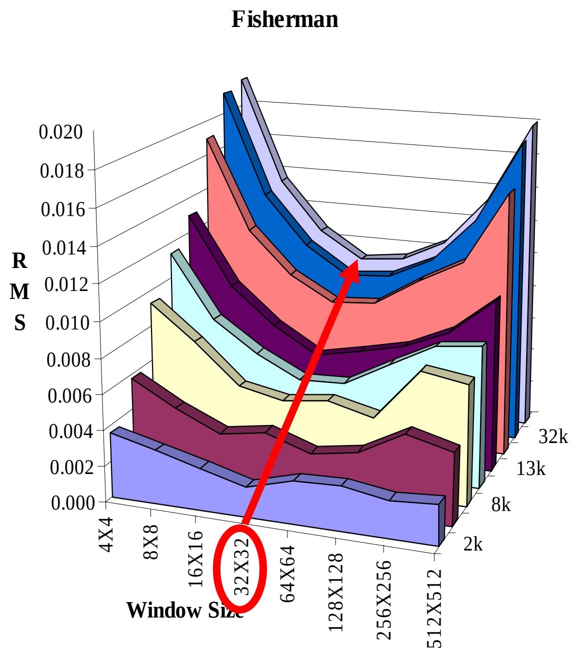


Figure 4. RMS vs. block size for the Fisherman 512x512 image

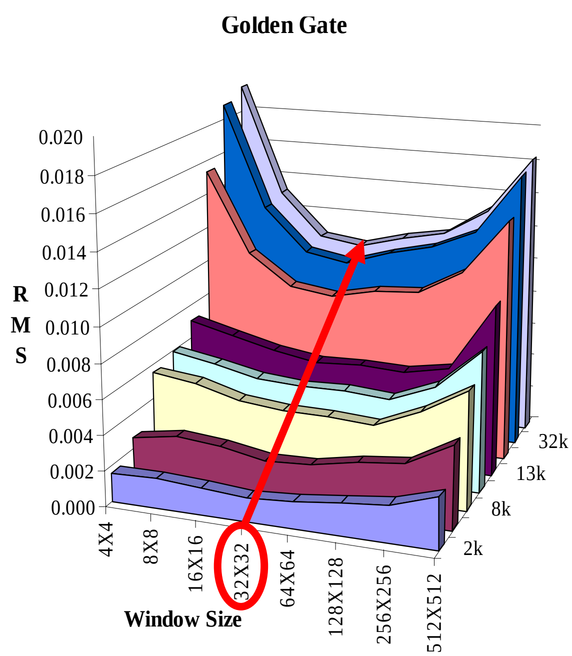
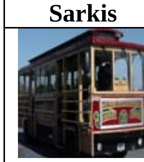


Figure 5. RMS vs. block size for the Golden Gate 512x512 image

4.3 Capacity Measure

Using the capacity measure we ranked the images. This measure was tested using over 100 images form our database. The analysis showed that the measure presented produced advantages over existing methods in all the test cases. As an example the two images in Table 3, "Rock" and "Fisherman", were selected as the best cover images.

Table 3: M= MSE* (Message Size/Cover-Image size)

Image	RMS	M	Cover Im
 Fisherman	0.0043	8.18E-07	2
 GoldenGate	0.0051	1.15E-06	4
 Sarkis	0.0044	8.56E-07	3
 Trolley	0.0051	1.15E-06	4
 Rock	0.0036	5.73E-07	1

4.4 Computer Simulation and Comparisons

In this section we test the presented algorithms and Fridrich's method [2] using the RS Steganalysis detection algorithm. In the results shown in tables 4 and 5 we can see the detection percentages per color layer and the detection's total percentage. For this analysis images were embedded with 95% and 19% of stego data. We can see that for the majority of the images, the steganalysis report shows fewer detection percentages when we used the presented embedding methods. In tables 1 and 2 it is also shown that the presented method introduces less distortion when embedding.

Table 4: This table contains the steganalysis report from RS steganalysis detection algorithm. The images were embedded with 95% of stego data.

Images	Red %	Green %	Blue %	Total %
Fisherman				
Clean	0.00	0.00	0.00	0.00
RGB(Fridrich)	0.00	6.05	11.14	5.73
RGB(Presented)	6.06	0.00	0.00	2.02
HSV(Presented)	0.00	3.67	0.00	1.22
YCbCr(Presented)	0.00	3.67	0.00	1.22

GoldenGate				
Clean	2.90	1.14	0.00	1.35
RGB(Fridrich)	15.01	10.97	0.00	8.66
RGB(Presented)	7.86	8.91	0.00	5.59
HSV(Presented)	8.71	4.97	0.56	4.75
YCbCr(Presented)	8.71	4.97	0.56	4.75
Rock				
Clean	4.20	0.00	0.00	1.40
RGB(Fridrich)	12.46	7.99	0.00	6.82
RGB(Presented)	6.50	0.00	0.00	2.17
HSV(Presented)	9.90	2.33	0.00	4.08
YCbCr(Presented)	9.90	2.33	0.00	4.08
Trolley				
Clean	0.00	3.71	0.00	1.24
RGB(Fridrich)	0.00	8.94	0.00	2.98
RGB(Presented)	0.00	0.26	0.00	0.09
HSV(Presented)	0.00	0.00	0.00	0.00
YCbCr(Presented)	0.66	0.00	0.00	0.22

Table 5: This table contains the steganalysis report from RS steganalysis detection algorithm. The images were embedded with 19% of stego data.

Images	Red %	Green %	Blue %	Total %
Fisherman				
Clean	0.00	0.00	0.00	0.00
RGB(Fridrich)	0.00	0.00	0.71	0.24
RGB(Presented)	0.00	0.00	0.00	0.00
HSV(Presented)	0.00	0.00	0.00	0.00
YCbCr(Presented)	0.00	0.00	0.00	0.00
GoldenGate				
Clean	2.90	1.14	0.00	1.35
RGB(Fridrich)	6.24	3.54	0.00	3.26
RGB(Presented)	4.39	2.54	0.00	2.31
HSV(Presented)	4.57	1.47	0.00	2.01
YCbCr(Presented)	5.81	1.00	0.00	2.27
Rock				
Clean	4.20	0.00	0.00	1.40
RGB(Fridrich)	6.48	0.95	0.00	2.48
RGB(Presented)	5.32	0.00	0.00	1.77
HSV(Presented)	5.90	0.00	0.00	1.97
YCbCr(Presented)	7.00	0.00	0.00	2.33
Trolley				
Clean	0.00	3.71	0.00	1.24
RGB(Fridrich)	0.00	5.05	0.00	1.68
RGB(Presented)	0.00	3.70	0.00	1.23
HSV(Presented)	0.00	0.92	0.00	0.31
YCbCr(Presented)	0.00	0.95	3.12	1.36

5. CONCLUSION

In this article, we presented a new windowing technique for embedding messages in palette/color-map based images. This new method has the advantage of embedding secure data, within the index, the palette or both, using special sorting scheme. The presented technique also incorporates the use color model and cover image measures in order to select the best of the candidates for the insertion of the stego information. Since this scheme embeds using MxN blocks and a weighted distance measure, it has the advantage that the manipulated colors will be closer. Therefore, it introduces lower distortion and higher embedding capacity than EzStego and Fridrich's method. In fact, computer simulations have shown that the presented algorithms carry with them the following advantages: 1) Introduces 8-10 times less distortion than EzStego [4] and 4-5 times less distortion than Fridrich's [1] method. 2) Provides additional security through a simple selective color and cover image algorithm. 3) Provides the closest color substitutes using a weighted distance measure. 4) Finally, it offers an increased capacity by embedding only on the best cover images.

6. ACKNOWLEDGEMENT

This research was partially funded by the Center for Infrastructure Assurance and Security under contract with the US Air Force (Air Force Research Laboratory), and Air Force Information Warfare Center.

7. REFERENCES

[1] Sos S. Agaian and Juan P. Perez, "New Pixel Sorting Method for Palette Steganography and Steganographic Capacity Measure," GSteg Pacific Rim Workshop on Digital Steganography, November 17-18, 2004, pp. 37-35, ACROS Fukuoka 1-1 Tenjin 1-chome, Chuo-ku, Fukuoka, 810-0001 Japan

[2] Jiri Fridrich, "A New Steganographic Method for Palette-Based Images", Center for Intelligent Systems, SUNY Binghamton, Binghamton, NY 13902-6000. IS&T PICS Conference, pages 285-289, 1999.

[3] Jiri Fridrich, "Secure Steganographic Methods for Palette Images", International Workshop on Information Hidding 1999, pp.47-60, 1999

[4] Machado, R.: http://www.fqa.com/ezstego/

[5] Michiharu Niimi, "High Capcity and Secure Digital Steganography to Palette-Based Images", University of Maine, IEEE ICIP pp. 917-920

[6] Niimi, Michiharu, "A BPCS Based Steganographic Method for Palette-Based Images Using Luminance Quasi-Preserving Color Quantization". Kyushu Institute of Technology. Pacific Rim Workshop on Digital Steganography, Japan.

[7] Kwan, M.: Gifshuffle. http://www.darkside.com.au/gifshuffle/