Adaptive Steganography based on textures

Dulce R. Herrera-Moro, Raúl Rodríguez-Colín, Claudia Feregrino-Uribe
National Institute for Astrophysics, Optics and Electronics
Luis Enrique Erro No.1, Sta. María Tonantzintla, Puebla, México C.P. 72840
{drosario,raulc,cferegrino}@inaoep.mx

Abstract

In this work we present a steganographic algorithm that allows an image process to identify the regions in a cover image where it is less probable to detect a hidden message using visual attacks. The regions selected are those whose texture is not homogeneous. This is because those kind of regions are originally noisy and is too difficult to detect extra information. Since the pixels where a message is hidden depend directly on the features of the cover image the steganographic system becomes adaptive. Tests were carried out with different kind of gray scale images and the results are compared against other approaches

1. Introduction

The term Steganography comes form the Greek words *stegos* (cover) and *graphy* (write). Therefore steganography literally means *covered writing*. Steganography unlike cryptography, does not change or scramble the message until it is illegible for an illegitimate receiver, it camouflages the message to occult its existence [4].

Steganography is an old method that has been used since the ancient Romans, mainly because it represent a secure mechanism of communication during wars. Nowadays, due to the boom of informatics, steganography uses digital media such as text, audio and video to hide information. Digital images are the most used way through the last years.

Steganalisys is the art and science of breaking the security of a steganographic system i.e. detecting the existence of a hidden message in a know medium. Most of the successful attacks have detected strong changes in the original medium caused by the presence of a hidden message. For this reason than adaptive steganography propose hiding information considering the features of the cover medium to identify the best regions to hide the data. A good place to hide data in digital images is a region with high contrast, several textures and many variations in its gray levels; because those regions generally are very noisy, and noise added for hiding a message is difficult to detect. In this work we present an algorithm which uses the statistical definition of texture in digital images to identify regions with local textures not homogeneous, from those regions we select some pixels to embed a message on their gray scale levels. The technique to hide information is LSB (least significant bit).

2. Attacks to Steganographic Systems

The steganographic attacks can be: *passive*, when the attacker is only able to analyze the information without changing it; or *active* when the attacker can manipulate the data. These attacks consist mainly in applying some sort of function that modifies the structure or intensity of an image to destroy a possible hidden message, although there is not evidence of its existence [2]. Based on the type of data available, the attack can be:

- *Stego-only*: if the attacker only has the stego-image.
- *Reuse of cover*: when a creator of steganograms uses the same cover to hide data more than once.
- *Know cover*: when a steganogram is intercepted and the attacker knows the original cover image.
- *Chosen stego*: when the attacker has the stego-image and knows the steganographic method used.
- *Chosen message*: in this method the attacker generates the stego-image with a know message in order to find signs to detect other stego-images.

The methods that we will present are mainly used against visual attacks. The visual attack method belongs to *passive only-stego* attacks and it is used to get the position and length of a message to extract it. Then, we will describe the main idea of this attack.

17th International Conference on Electronics, Communications and Computers (CONIELECOMP'07)
0-7695-2799-X/07 $20.00 © 2007 IEEE

Since the steganographic algorithms insert the information in the least significant bit (LSB) visual attack consists of remove all the bits that does not occult the message and keep only the least bits, then those bits are display in the following way: those bits whose value is 0 are displayed in black and those bits whose value is 1, are displayed in white. Visual attacks work well in images where the message is inserted in homogenous regions showing the position of the message as noise.

3. Image Processing for Adaptive Steganography

The Steganographic algorithms use a key and a embedding function to determine the position of the message into the image. To achieve the adaptive Steganographic algorithm we consider the following features of the embedding function [1].

- Pixel selection for embedding the data.
- The bit representation of the message.
- Modification of the cover data

The adaptability in the algorithm will occur in the selection of the pixel used for embedding, first we determine the regions where the noise is added by embedding a message is difficult to detect; these regions have a high standard deviation in its grayscale levels. The image obtained can be used in the Steganographic process using two schemes:

In the first scheme the processed image is used directly like a container of the message, in the second scheme a template is used to determine the position in the original image where the information will be hidden. Figure 1 shows those schemes.

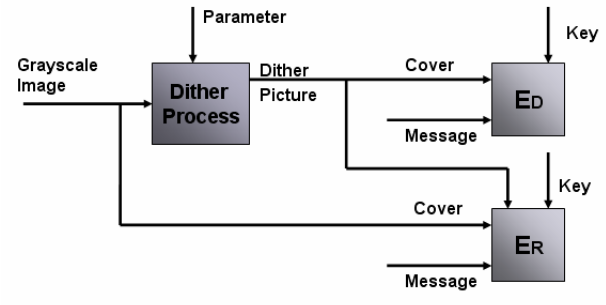


Figure 1. Possible Steganographic Systems using image

The algorithms presented in this work used the second scheme. The dithering process is used in the first two algorithms and texture detection is used in the third one.

3.1. Dithering

Dithering is a color reduction method used by monochrome monitors and printers. The color reduction objective is to simulate new colors when only a few basic colors are available. Dithering method is able to represent images in grayscale using only two colors: black and white.

There exist different dithering methods for grayscale image representation; nevertheless the basic idea is processing original image pixels with some method and later on to compare the result with a threshold. Given a pixel in the result image if this exceed the threshold then the white color is assigned to its corresponding pixel in the dithered image, otherwise black color is assigned.

3.1. Texture

Texture can be defined as a repetition of a patterns along a region, these patterns are formed by elements with specific features like size, form, color and direction.

There are three sort of methods to detect and analyze digital image textures:

- *Spectral methods*. Use Fourier transform to detect textures with periodic patterns, i.e. the prominent tips of the image spectrum shows the main patterns direction whereas the tips in the frequency plane shows the spatial period patterns.

- *Structural methods*. Define texture as a structure that is repeating along the region. That structure is called primitive texture. The object of this kind of methods is to find the primitive texture to study its distribution.

- *Statistical methods*. Use measures obtained from the pixel distribution values to characterize textures.

Statistical methods are employed by the proposed algorithms in this work

4. Used Algorithms

In this section three steganographic methods are described, these algorithms analyze the image features before embedding to determinate feasible regions to hide the data. The first two algorithms where proposed in [1] were called ConDith and ConDithSpread respectively.

These algorithms are based on dithering process to determinate feasible regions for embedding.

The third algorithm is based on the algorithm presented in [4] and uses the definition of texture proposed by the statistic methods to detecting regions with textures not homogeneous. We called this method *Context.*

In next subsections we will describe how each algorithm works.

4.1. ConDith

In this algorithm a region is selected if it contains a high degree of randomness in its grayscale distribution considering the local contrast as dithering criteria.

Therefore the algorithm calculates the difference between a pixel and its neighbors. If the difference exceeds a minimum threshold C_{min} then pixel can be used to embed a message bit and its corresponding pixel in the dithering image is marked with 255, otherwise it is marked with 0.

pixel[*x,y*] is usable if dither [*x,y*]=255

$$Pixel[x,y] = \begin{cases} 0 : diff > C_{\min} \\ 255 : otherwise \end{cases}$$

Where *diff* is the difference of pixel[*x,y*] to all adjacent pixels

4.2. ConDithSpread

This algorithm [1] is a modification of *ConDith* that increases the randomness of the selected pixels by taking only a few pixels of the selected by *ConDith*. The selection criterion is as follows:

pixel[x,y] is usable if dither [x,y]=255

$$Pixel[x,y] = \begin{cases} 0 : diff > C_{\min}, rnd[x,y] < \dfrac{(n+1)}{2} \\ 255 : otherwise \end{cases}$$

where *rnd* is a random number in $[0,n]$.

4.3 ConText

To identify those regions we analyze the grayscale space distribution of a region to select the areas with greater diversity of grayscale levels. The pixel selection process is the following:

- Divide the image in non overlapping block of size 3x3 pixels.
- Divide each block in four sub-blocks (Figure 2.).
- Each sub-block is *good* if there are at least three different grayscale levels. (See Figure 2 b-e).
- Select the central pixel if the four sub-blocks are *good* before and after embedding.

In Figure 2 an example of this process are showed.

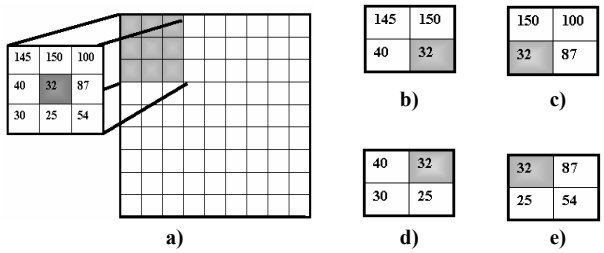


Figure 2 Example of pixel selected in ConText

Figure 3 shows an example of results obtained using the three algorithms: a) is the original cover image b) shows in gray boxes the pixels where the message bits can be embedded using *ConDith* algorithm, c) shows the pixels selected using *ConDithSpread* algorithm and d) using the proposed ConText algorithm.

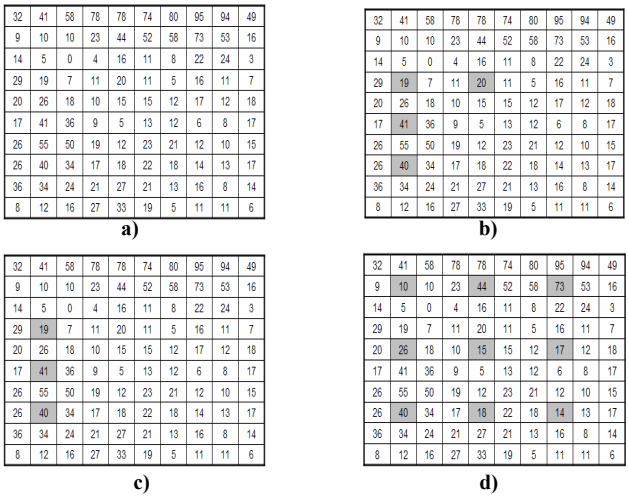


Figure 3. Pixels selected using the three algorithms. a) Original image. b) ConDith c) ConDithSpread d)Context.

0-7695-2799-X/07 $20.00

As we see in Figure 3 the three algorithms idea follow the same principle of occult the information in areas where the noise caused by insert a message can not be detect. However, the difference between them is the way to select those areas.

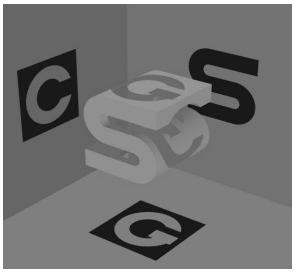
Context unlike Condith or ConditSpread does not chose regions where exist great differences between tones of the pixels. Context selects regions where exists a great variety in the tones, this allows obtaining more places to embedding when the image have poor contrast but exist not homogenous surfaces.

5. Evaluation and Results

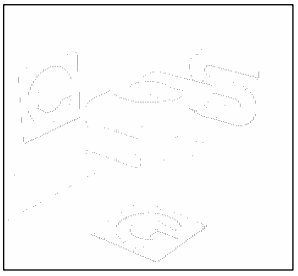
In this section we have two main objectives, the first is to show how the three algorithms that was presented in section four can use the information of the cover image to determinate which are the best regions to insert the message without be detectable using and visual attack.. That fact made the algorithm adaptable increases the security over LSB

The second objective is show how the criteria used for Context can increases the capacity obtained for ConDith in images which don't have many contrast but have regions with enough different values to hide information.

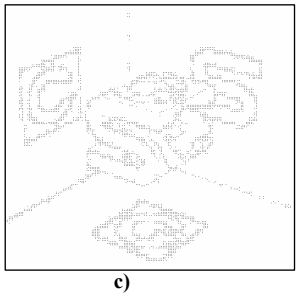
The experiments were made with bmp gray scale images of size 512x512. From figure 4 to 7 show the position of pixels that were chose to insert information according to ConDith and ConText. They are present like black points in a template. The pixels selected by ConDithSpread are not presented because they were a few.



a)

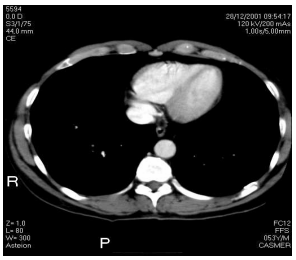


b)

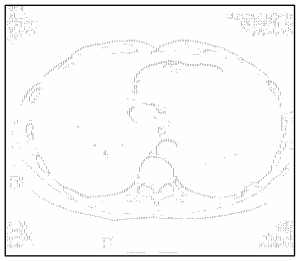


c)

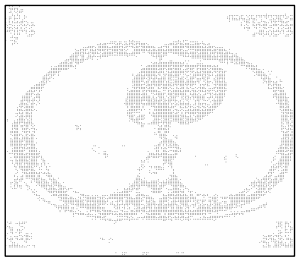
Figure 4 Bytes selected to insert information a) Original image. b)Pixels selected by Dithering c) Pixels selected by Context.



a)

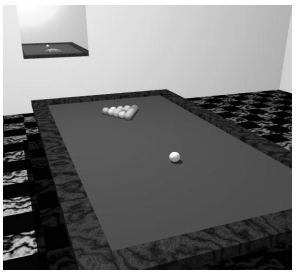


b)

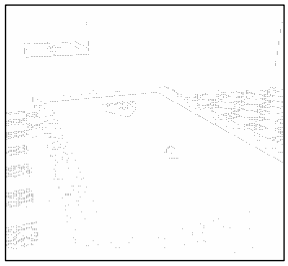


c)

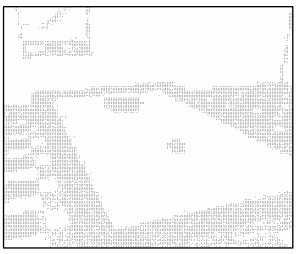
Figure 5 Bytes selected to insert information a) Original image. b)Pixels selected by Dithering c) Pixels selected by Context.



a)

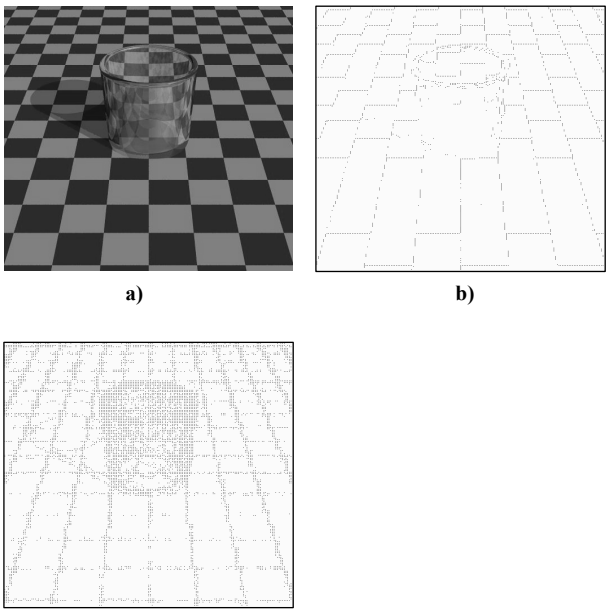


b)



c)

Figure 6 Bytes selected to insert information a) Original image. b)Pixels selected by Dithering c) Pixels selected by Context



a) b) c)

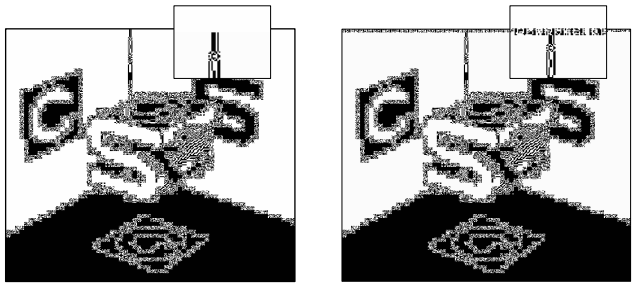
Figure 7 Bytes selected to insert information a) Original image. b)Pixels selected by Dithering c) Pixels selected by Context

In the table1 we present the numbers of bits maximums that can be use on the images showed before. In each image ConText obtain major capacity.

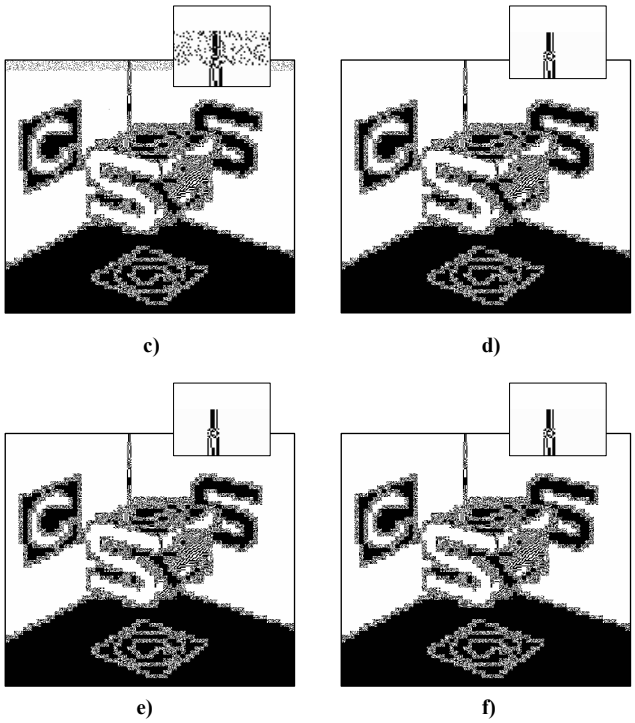
IMAGE	CONDITH SPREAD	CONDITH	CONTEXT
Csg	323 bits	654 bits	3740 bits.
Pool	742 bits	1516 bits	8693 bits
Ultra	1057 bits	2220 bits	7164 bits
Glass	1082 bits	2148 bits	7988 bits

Table 1 Maximum capacity obtained reached by each algorithm .

Figure 8 shows visual attack to image Csg image. All of bits available for each method were use to insert information except in LSB and LSBSrpread that inserts the same numbers of bits as Context.



a) b)



c) d) e) f)

Figure 8 Visual Attack. a) Originals bits of the image b) LSB c)LSBSpread. d)ConDith. e) ConDith Spread f) Context .

Figure 8 shows that using adaptive algorithms the message is not detected with visual attacks.

6. Conclusions

In this work we presented a steganographic algorithm called Context that selects pixels to embedding information from non homogeneous texture regions. This algorithm was compared with the algorithms presented in [1] called ConDith and ConDithSpread. The major contribution of this kind of algorithms is the selection of pixels for embedding based on the features of the cover image. This reduces the probability of detection.

The criterion used in the Context to select the pixels for embedding does not allows the message to be detected by visual attack and allows major capacity of embedding that with ConDith and ConDithSpread in images with poor contrast.

The major Drawback of the three algorithms is that after the selection of pixels, they use LSB to insert the message and if we apply any filter in the stego-image the message is lost. In order to improve the algorithms we can change the insertion method.

7. References

[1]A. Franz, A. Schneidewind, "Adaptive Steganography Based on Dithering", Proceedings of the 2004 Workshop on Multimedia and Security, Magdeburg, Germany, 2004, pp. 56-62.

[2] I. Avcibas, N. Memon and B. Sankur, "Steganalysis Using Image Quality Metrics", IEEE transactions on image processing, Vol. 12, No.2, February 2003, pp.221-229.

[3] J. Fridich, "Secure Encryption and Hiding of Intelligence Data", Dept. of Defense, Air Force Rome Laboratory Phase 1, Final Technical Report,1998.

[4] J. Fridich, and R. Du, "Secure Steganographics Methods for Palette Images", In Information Hiding, 3rd International Workshop, Springer 1999,pp. 47-60.

[5] B. M. Planitz, A. J. Maeder, "Medical image watermarking: a study on image degradation". In Lovell, B. C. and Maeder, A.J., eds. APRS Workshop on Digital Image Computing (WDI2005); Griffith University. 2005: 3-8 CD ROM. ISBN: 0958025533.

0-7695-2799-X/07 $20.00