American Journal of Applied Sciences 4 (9): 658-662, 2007
ISSN 1546-9239

Colored Digital Image Watermarking using the Wavelet Technique

[1]Mohammed F. Al-Hunaity, [2]Salam A. Najim and [2]Ibrahiem M. El-Emary
[1]Department of Information Technology, Prince Abdullah Bin Ghazi Faculty of Science and Information Technology, Al-Balqa' Applied University, Al-Salt 19117, Jordan
[2]Department of Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman 11910, Jordan

Abstract: With the revolution of information technology and Wide Area Networking, data has become less and less private where the access of media as well as the attempts to change and manipulate the contents of media data have become a common case. For that, we need to use a watermarking technique to protect the copyright of the media as well as for digital right management but without leaving a visual effect. We presented a watermarking technique that deals with images where the used technique to embed a wavelet compressed watermark image within the least significant bit (LSB) of the cover image pixels in a specific pattern which won't be visible after embedding and will cause the cover image to become copyrighted using the embedded watermark image that can be extracted later.

Keywords: Watermarking, Steganography, Wavelet, Compression, LSB, Image, BMP, RGB.

INTRODUCTION

Steganography means hiding messages in innocuous carrier to try to conceal the existence of the message while watermarking is almost the same as Steganography but the only difference between them is the purpose of hiding the data within the cover image, video, audio or text [1]. Steganography is usually used to hide data for reasons of delivering hidden and secret messages to the other end.

The objective of the watermarking is to hide data for digital right management and copyright protection of the digital image, video or audio that holds the secret watermark. There are a lot of the watermarking applications such as: broadcast monitoring in which the watermark is embedded in the advertising spot by the advertisers. In this way, we will know if another station pirated the advertisement that has their watermark. Copy control, while some companies allow recording TV broadcast, they don't allow recording another copy of the broadcast and that is accomplished by adding a fragile watermark that says (copy allowed) which will not appear if a video recorder is used. Another application is Content authentication which is done by adding a watermark to digital works, photographs, surveillance camera videos as well as important scanned documents [2].

With the global and wide use of internet and different network topologies, different data and media types have become less and less protected since anyone could download these data or media and modify it without anyone knowing about it. Because of that, many copyright problems appeared lately and different watermarking techniques were proposed; one of them is by adding a visible watermark to the cover image while the other one is to use a watermarking techniques that will embed a hidden watermark within the cover, where the watermark is the message that will be embedded into the pixels of the image cover, while the cover image is the image that will be watermarked using the watermark image to protect it from being claimed by other persons or groups. The result of this watermarking process is the Stego image which is the cover image after being watermarked.

In this paper, some previous works will be mentioned in the related work section. The problem of the non watermarked images will be discussed in the Problem definition section. After that, the method used to achieve the watermarking is mentioned in details in Methodology section while output image and SNR results will be shown in Output results section. Summary of the paper, the conclusion as well as the future work will be discussed in the final section. Finally the references used will be mentioned in a reference list at the end of the paper.

Corresponding Author: Ibrahiem M. El-Emary, Department of Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman 11910, Jordan

The use of encryption exist for a while now since it is used to encrypt regular data, images, text, or anything into another form, whom only the two parts of communication having the keys for decryption. But as tempting as it seems, encryption is still considered a way of hiding information and someone could be accused of shady or illegal activities. Another alternative was found which Steganography is, and using it almost anyone who can hide data in the plain sight and the most used mediums for that are graphics and audio since the small undetected change won't draw attention. There are lots of programs that are capable of achieving the hiding process on graphics or audio such as: JP hide and JP seek which are programs that hide data after encrypting within a jpeg; MP3 Stego which is a program that compress, hide and encrypt data within an MP3 file; Sam's Big Play Maker which hides a text within a play and S-Tools which is a program that can hide in WAV, audio file, BMP files, or unused space in a floppy disk [3].

Digital Steganography has been found for a while now and it has been used by lot of different ways; it was used in hiding text within a text like using the first letter of each word and makes words of these letters. Another way is using hypertext comment notation to hide data inside it or hiding text or data inside an audio file, or in an image or video. Here, the main idea is hiding the data in the redundant bits where the change of the cover nature is non noticeable. In Images, the colored images (24 bits/pixel) that consist of RGB values of colors are a better environment for embedding data than the grey scaled images (8 bits/pixel), and usually data are hidden in the redundant bits which are usually the least significant bits of the image since their change won't be noticeable by the human eye system . Some images are considered better than others for using as: a cover image, images that have variances in colors and don't have much of plain color areas are better covers for embedding and hiding data into [4]. Using RGB images (24 bits/pixel), we can achieve high storage capacity for embedding data into and a good way to do that is to embed the data into the two least significant bits of each color band in the pixel value. This way will use the least two bits of the Red color, the least two bits of the Green color, and the least two bits of the Blue color of each pixel to get 6 bits out of 24 bits for hiding the message. This technique is also useful because it can survive some of the Lossy compression algorithms making the Steganography more robust against the different image processing techniques [5].

Problem Definition: Lots of the images that are available in the internet nowadays are not watermarked, so this will make it is very easy for any one to download these images and modify them, and this could cause the loss of original creator's rights of this image. Accordingly, the proposed solution is to embed a watermark image within the pixels of the cover, but still there is another problem, when an image is being embedded, it shouldn't cause any visual change to the cover, that's why LSB insertion should be used but another problem appears with this since the image is limited by it's dimensions, the number of bits that are usable for embedding is also limited and the watermark image should be chosen in such that it could fit in the cover; here comes the rule of the wavelet compression to reduce the size of image.

Proposed Methodology of Solution: The proposed method in this paper is to use a watermark image which has the dimensions of 512×512 gray scale and will be embedded into a cover image having dimensions of 1024×768 RGB where the watermark image (message) will be wavelet Haar level 2 compressed then embedded into the bits of the cover image.

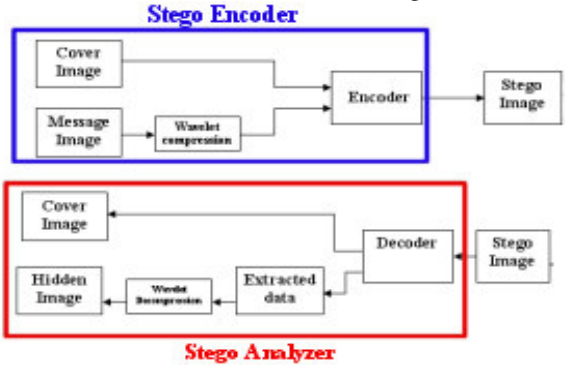

Fig. 1: The Watermarking System structure diagram

The watermark image will be wavelet Haar level 2 transformed, then the difference coefficients of the wavelet transform will be quantized into 3 bits each, where they were 8 bits each. The result of the transform will be as follows:

Fig. 2: Cathedral [6]

Wavelet Haar Transform level 2

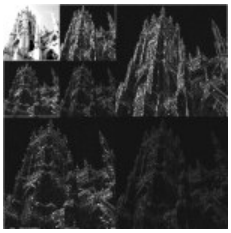

Fig. 3: Wavelet Haar level 2 transform

The average coefficients is the upper left light part of Fig. 3, where the other parts are the difference coefficients which will be quantized into 3 bits each, while the average coefficients will stay 8 bits each. After that, threshold will be used to reduce the error later at extraction, since negative difference coefficients values can exist, and threshold works as follow: If $S_i < Thr$, then $S_i = 0$, where S_i is the pixel (I) in the transformed image. All the previous steps were applied on the watermark gray image, now the watermark image is ready to be embedded into the bits of the cover image pixels. The embedding is done as by embedding region after region, where each region of the watermark image will be embedded into a region in the cover image.

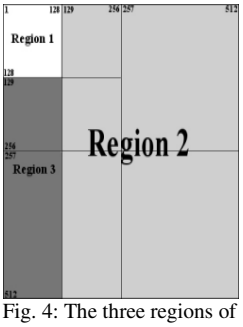

Fig. 4: The three regions of of the cover image

Embed

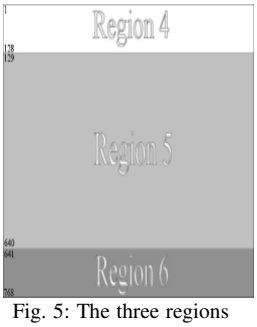

Fig. 5: The three regions the watermark image

Embedding Region 1 (rows 1-128, columns 1-128) into Region 4: Each three pixels of the message will be embedded into four corresponding pixels in the cover in the following manner as shown in Fig. 6, where the rows will be embedded one by one.The 8 bits of the each three pixels of the message will be embedded in the 2 least significant bits of each color of each pixel for the four corresponding pixels, where each pixel in the cover consists of red byte, green byte and blue byte. Region 1 of message will be embedded into region 4 in cover image.

The sequence of pixels is as follows:

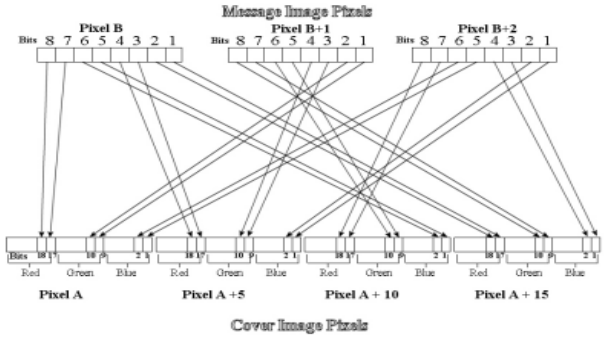

Fig. 6: The Algorithm used to embed each 3 pixels in region 1 of watermark image into each 4 pixels from Cover image

For j = 1 To 128
For i = 1 To 43
Message pixel (1 + 3 * (i - 1), j)
Message pixel (2 + 3 * (i - 1), j)
Message pixel (3 + 3 * (i - 1), j)

// for odd rows (odd j) //
Cover Pixel (101 + 20 * (i - 1), j)
Cover Pixel (106 + 20 * (i - 1), j)
Cover Pixel (111 + 20 * (i - 1), j)
Cover Pixel (116 + 20 * (i - 1), j)

// for even rows (even j) //
Cover Pixel (103 + 20 * (i - 1), j)
Cover Pixel (108 + 20 * (i - 1), j)
Cover Pixel (113 + 20 * (i - 1), j)
Cover Pixel (118+ 20 * (i- 1), j)

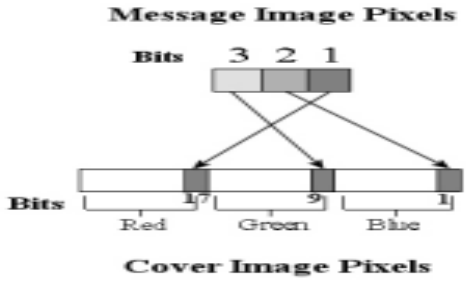

Fig. 7: The algorithm used to embed each pixel of region 2 of the watermark image into each pixel in the cover image

Embedding Region 2 (rows 1-512, columns 129-512) into Region 5. Each pixel of the message image will be embedded into the least significant bits of each color of cover pixels, as show in Fig. 7 Each pixel of the message is 3 bits now these are the quantized coefficients. The sequence of pixels is as follows: Each

row of message will be embedded in a row of cover.

```
For j = 1 To 512
For i = 129 To 512
Message Pixel (i, j)
\\ Odd columns ( odd j )
Cover Pixel (1 + 2 * (i - 129), j + 128)
\\ Even columns ( even j )
Cover Pixel (256 + 2 * (i - 129), j + 128)
```

Embedding Region 3 (rows 129-512, column 1-128) into Region 6: In this region embedding, the same pixel embedding algorithm is used as in the previous one, but the sequence of pixels is different. Here each column of the message image will be embedded into each row of the cover image. The sequence of pixels is as follows:

```
For i = 1 To 128
For j = 129 To 512
Message Pixel (i, j), 3)
\\ Odd columns (odd i )
Cover Pixel (1 + 2 * (j - 128), 640 + i)
\\ Even columns (even i )
Cover Pixel (256 + 2 * (j - 128), 640 + i)
```

This way, pixels of the compressed message image are embedded into the cover image.

OUTPUT RESULTS

The result of this system varies due to the chosen threshold value. These are the extracted watermarks with the different threshold values:

Fig. 8: Original watermark Image [7]

Fig. 9: The Extracted watermark with threshold = -100

Fig. 10: The Extracted watermark With threshold = 0

Fig. 11: The Extracted watermark with threshold = +100

Here is the result of embedding the previous watermark into the cover image:

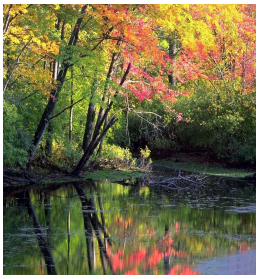

Fig. 12: The Original Cover Image [8]

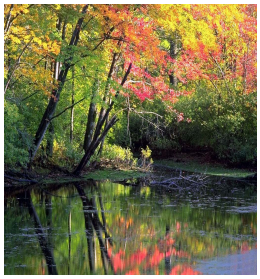

Fig. 13: Watermarked Image (Stego Image) using threshold = 0

We can see that there isn't a visual difference between the two images, the original cover image, and the watermarked image (Stego Image) due to LSB insertion. For the previous embedded watermark into this cover image, with the different threshold values, the followings are the different SNR values. We can find from these results that the best case happens when the values of the threshold is in the range (-5 to +5).

Threshold	-100	-15	-5	0
PSNR	-63.1365	-63.1364	-63.136	-63.136

Threshold	0	+5	+15	+100
PSNR	-63.136	-63.136	-63.1364	-63.2896

CONCLUSION

Digital image watermarking is important to all kinds of media, to keep them from being claimed by other non related people, or by being edited or modified. From the results of the embedding and extraction, we can find that the best values for getting the best image compression is using the threshold values from (-5 to 5). The wavelet compression helped in reducing the size of the watermark image, and therefore reducing the effect of embedding the watermark image into the cover. The current system watermarked image, can survive both TIFF and TGA compression without affecting the hidden

watermark. The system still doesn't survive JPEG compression; this could be handled in future work.

REFRENCES

1. Stefan, K. and F. A. Petitcolas (Editors), 2000. Information hiding techniques for steganography and digital watermarking. Artech House Books.
2. Ingemar c., M. L. Miller and J. A. Bloom, 2001. Digital watermarking, Morgan Kaufmann publisher.
3. http://opcug.ca/public/Articles/0112.PDF
4. http://druid.caughq.org/presentations/Steganography-Primer.pdf
5. http://csrc.nist.gov/nissc/1996/papers/NISSC96/paper014/STEGOX.PDF
6. http://www.cis.nctu.edu.tw/~whtsai/EnglandTrip/9th%20Day/pictures/02%20York/04%20York%20Minster%20C.athedral%20---%20Outside%20architecture%20E.jpg
7. http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/lena.html
8. http://wp.li.ru/natura