Image Watermarking based on Huffman Coding and Laplace Sharpening

[*]Muna M. Lafta

Abstract

In this paper, an algorithm through which we can embed more data than the regular methods under spatial domain is introduced. We compressed the secret data using Huffman coding and then this compressed data is embedded using laplacian sharpening method.

We used Laplace filters to determine the effective hiding places, then based on threshold value we found the places with the highest values acquired from these filters for embedding the watermark. In this work our aim is increasing the capacity of information which is to be embedded by using Huffman code and at the same time increasing the security of the algorithm by hiding data in the places that have highest values of edges and less noticeable.

The performance of the proposed algorithm is evaluated using detection techniques such as Peak Signal- to- Noise Ratio (PSNR) to measure the distortion, Similarity Correlation between the cover-image and watermarked image, and Bit Error Rate (BER) is used to measure the robustness. The sensitivity against attacks on the watermarked image is investigated. The types of attacks applied are: Laplacian sharpening, Median filtering, Salt & Peppers Noise and Rotating attack. The results show that the proposed algorithm can resist Laplacain sharpening with any sharpening parameter *k*, besides laplacian good result according to some other types of attacks is achieved.

1-Introduction

With the increasing use of internet and effortless copying, tempering and distribution of digital data, copyright protection for multimedia data has become an important issue. Digital watermarking emerged as a tool for protecting the multimedia

[*] Baghdad University, Collage of Education for Women, Computer Science Department.

data from copyright infringement. In digital watermarking an imperceptible signal "mark" is embedded into the host image, which uniquely identifies the ownership. After embedding the watermark, there should be no perceptual degradation. These watermarks should not be removable by unauthorized person and should be robust against intentional and unintentional attacks [1].

Many researches have been spending during the last years to develop image watermarking schemes. Lalitha [2] proposed image watermarking based on Huffman coding using modified auxiliary carry. Jun [3] proposed a semi-fragile watermarking tolerant of laplacian sharpening by modifying the parity of pixel.

In this paper, an image watermarking method based on Huffman coding and laplacian sharpening is proposed, by using them increased security and the robustness as explained in section result.

2. Huffman Coding:

Secret data it will be encoded by Huffman coding**.** Huffman coding is an entropy encoding algorithm used for lossless data compression developed by David A. Huffman. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It can be informally defined as a prefix-free binary code (a set of code words) with minimum expected codeword length (equivalently, a tree with minimum weighted path length). Formally it can be defined as follow [2]:

Input: (i) Let the input be an array of alphabet A= {a1, a2, …..an},which is the symbol alphabet of size n.

(ii) Let W= {w1,w2,……wn}, which is the set of the positive symbol weights Wi=weight(ai), $1 \leq i \leq n$

Output: Code C(A,W) = {C1,C2,…..Cn}, which is the set of (binary) code words, where Ci is the codeword for ai, $1 \leq i \leq n$.

To illustrate the work of Huffman coding assume the string "go go gophers" is encoded in ASCII, how we might save bits using a simpler coding scheme, and how Huffman coding is used to compress the data resulting in still more savings. With an

ASCII encoding (8 bits per character) the 13 character string "go go gophers" requires 104 bits. The table below shows how the coding works.

Table 1: ASCII and 3-bit coding

ASCII coding **3-bit coding**

char	ASCII	binary	char	code	binary
g	103	1100111	g	0	000
o	111	1101111	o	1	001
p	112	1110000	p	2	010
h	104	1101000	h	3	011
e	101	1100101	e	4	100
r	114	1110010	r	5	101
s	115	1110011	s	6	110
space	32	100000	space	7	111

The string "go go gophers" would be written (coded numerically) as 103 111 32 103 111 32 103 111 112 104 101 114 115. Although not easily readable by humans, this would be written as the following stream of bits (the spaces would not be written, just the 0's and 1's)

1100111 1101111 1100000 1100111 1101111 1000000 1100111 1101111 1110000 1101000 1100101 1110010 1110011

Since there are only eight different characters in "go go gophers", it's possible to use only 3 bits to encode the different characters. We might, for example, use the encoding in the table on the right above, though other 3-bit encodings are possible.

Now the string "go go gophers" would be encoded as 0 1 7 0 1 7 0 1 2 3 4 5 6 or, as bits:

000 001 111 000 001 111 000 001 010 011 100 101 110 111

By using three bits per character, the string "go go gophers" uses a total of 39 bits instead of 104 bits. [5].

3. Laplacian Sharpening

The Laplacian sharpening based on the theory of laplacian operator, an image watermarking algorithm will be proposed to distinguish Laplacian sharpening from other image operations.

Laplacian operator is one of the commonly used image sharpening operators, it can be described by the following formula

$$\nabla^2 f(i,j) = f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1) - 4f(i,j) \quad \ldots\ldots (1)$$

Where $f(i, j)$ represents the pixel value of a digital image. According to (1), The Laplace formula simply measures the difference between a pixel and its four touching neighbours. When Laplacian operator is used to get the sharpened image, the sharpened pixel n value is

$$g(i,j) = f(i,j) - k\nabla^2 f(i,j) \quad \ldots\ldots (2)$$

$$g(i,j) = 4kf(i,j) + f(i,j) - k[f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1)] \quad \ldots\ldots (3)$$

Where k is a parameter concerning with diffusion effect, and it should be chosen reasonably according to the practice situations, since it affects the sharpening effect directly. With an appropriate k, Laplacian operator can be used to get a clear-cut image from blurry a image, while the content of the image remain unchanged, so it is widely used in digital image processing. The pixel value of the image is changed by sharpening, but the content of the image is unchanged [3, 4].

4. The Proposed Method

In this paper we have proposed an algorithm in which we used the principle of Huffman coding for encoding the secret data and the theory of Laplacian sharpening, which can distinguish laplacian sharpening from other attacks. The encoded string is embedded into cover image after applying Laplacian sharpening, then based on threshold values we found the places with the highest value produced from filter, use these places for embedding the watermarked. we used laplacian formula for two main reasons first, the values produced by it use as index in the original image, so it provide the randomness, second watermarked images often contain small variations, and can be detected easily by examining Laplace magnitude counts. Therefore it seems reasonable to suggest if a hiding algorithm were able to use the Laplace formula

during embedding, it would be able to hide the information in a less noticeable way.

Figure (1) presents the main steps of the embedding stages of the proposed algorithm. The extracting module consists of the corresponding stages which perform the inverse tasks of those done in embedding stages. In the following, the embedding and extracting methods are introduced.

4.1 The watermark embedding process steps:

Step1: The first stage in the suggested system is loading the bitmap data from image and the file that contain the secret massage want to hide.

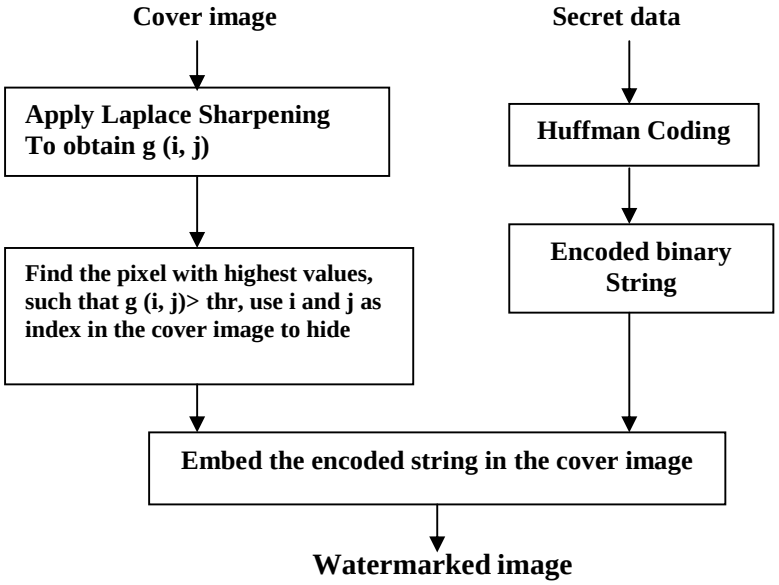


Figure 1: The scheme of the proposed methodology

Suppose the cover image is a color image with size of $M \times N$, and the watermark is a binary bit stream with the length of L, where $L \leq M \times N / 9$.

Step 2: Encode the secret message using Huffman coding.

Step 3: Divide the image into blocks of 3 × 3 pixels, thus the cover image is divided into blocks with size of 3×3 in order to make full use of the basic theory of Laplacian sharpening.

Step 4: Compute the Laplacian sharpening result of the central pixel in every block according to (2) that is

$$g(n,2,2) = f(n,2,2) - k\nabla^2 f(n,2,2) \quad \ldots\ldots (4)$$

Step 5: Based on threshold values, find the places that have the more changes in values (highest values). This done by comparing the pixels output from laplacian

sharpening with threshold value. If g (i, j) > threshold, store (i, j) use them as index in the cover image to embed the binary stream produced from Huffman encoder.

Step 6: Embed the binary stream in cover image using LSB. The formula for the embedding is as follows:

$$x' = x - x \bmod 2^{k} + b \qquad \text{..... (5)}$$

Where k is the number of LSBs to be substituted.

4.2 The extracting process steps:

The extraction algorithm is the reverse process of the embedding algorithm. The steps are summarized as follows:

***Step1*:** Apply laplacian sharpening on the watermarked image.

***Step2*:** Obtain the index of highest values.

***Step3*:** Extract the embedded bits from places with these indices

$$b = x \bmod 2^{k} \qquad \text{..... (6)}$$

***Step4*:** Decode the secret message into original message by decoding Huffman code.

5. Objective Tests

Since the essential goal of steganography is concealing the fact that a secret message is transmitted, then it is very important to make the stego-image to be as close as possible to the cover-image. In this section, two objective tests are presented to measure the security of the considered stego-system.

5.1 The Similarity Test

Similarity test is the correlation between the cover-image and stego-image. When the stego-image is perceptually similar to the original cover-image, then the correlation equals one. The correlation can be calculated using [6]:

$$Cor = \frac{\sum_{r=1}^{M}\sum_{c=1}^{N}(C(r,c)-\overline{C})(S(r,c)-\overline{S})}{\sqrt{\left[\sum_{r=1}^{M}\sum_{c=1}^{N}(C(r,c)-\overline{C})^{2}\right]\left[\sum_{r=1}^{M}\sum_{c=1}^{N}(S(r,c)-\overline{S})^{2}\right]}} \qquad \text{..... (7)}$$

where:

r: row number, c: column number, M: height of cover image (or stego-image) ,N: width of cover image (or stego- image) , $C(r,c)$: cover image , $S(r,c)$: stego-image , $\bar{C}$: mean of cover image , $\bar{S}$ mean of stego-image

$$\bar{C} = \frac{1}{MXN} \sum_{r=1}^{M} \sum_{c=1}^{N} C(r,c) \quad \text{..... (8)}$$

$$\bar{S} = \frac{1}{MXN} \sum_{r=1}^{M} \sum_{c=1}^{N} S(r,c) \quad \text{..... (9)}$$

5.2 Peak Signal- to- Noise Ratio (PSNR) Test

According to the human visual system, some amount of distortion between the original image and the modified one is allowed. Here, the *PSNR* is employed to indicate the performance of the method. *PSNR* is usually measured in *dB* [7].

PSNR = 10 log10 (255^2/ MSE) ..... (10)

MSE= (1/N) ^{2} **ΣΣ** (xij – x'ij) ^{2} ..... (11)

Where x_{ij}'s denote the original pixel values, x_{ij}' denote the modified pixel values, and N is the modified dimension of image.

5.3 Bit Error Rate (BER)

The Bit Error Rate Calculation compares input data from a transmitter with recovered data from a receiver:

$$BER = \frac{Error\_bits}{Total\_Secret\_bits} \quad \text{..... (12)}$$

The BER is of great significance to system design, since it is strongly related to practical system requirements such as robustness [7].

6. Results

The proposed method is evaluated in three color images (vary in size and in details):"Computer", Brides" and "Rose", the size of each image explained in Table 2. The performance measures are the invisibility of the inserted watermark and the robustness of the method against various types of attacks. In the experiment, the peak signal to noise ratio (PSNR) is used to measure the embedding distortion, Similarity is used to measure correlation between the cover-image and watermarked image and (BER) is used to measure the robustness.

Figure (1) shows the original images and the watermarked images. It is obvious that the watermarked copy is undistinguishable from the original image. It is

evident that most of the watermark data are added to the edges where they are perceptually invisible. Figure (2) shows the cover images after laplacian sharpening and the highest values for these images. Figure (3) shows the secret data want to embed and Huffman coding to it, the length of secret data before compression (840 Bits) and after Huffman coding (380 Bits).

Table2 depicts the objective quality values of the proposed method for the tested images. The attacks employed for testing are Laplacian sharpening, median filtering, Salt & Peppers noise and rotating the image by 20°. The BER under Laplacian sharpening with various k is shown in figure 4; it indicates that the proposed algorithm can resist Laplacian sharpening with any sharpening parameter k. Besides Table 3 BER under other types of attack explained. Figure 5 shows watermarked images after attack.

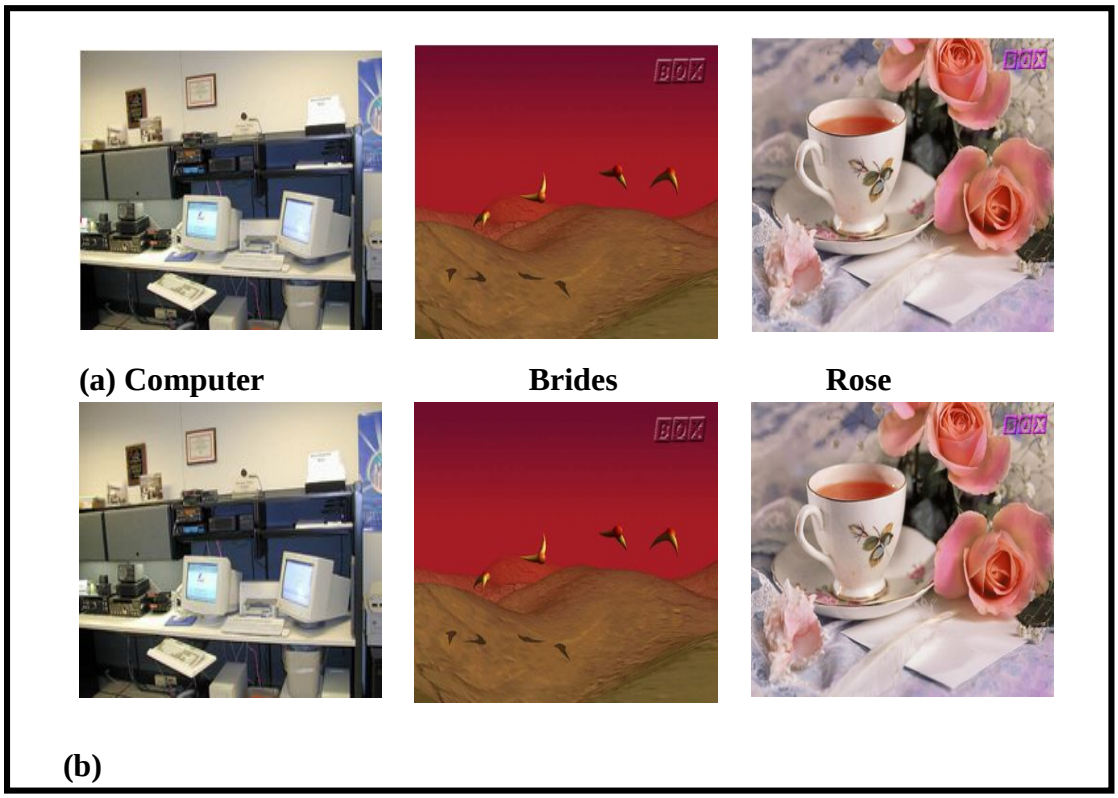
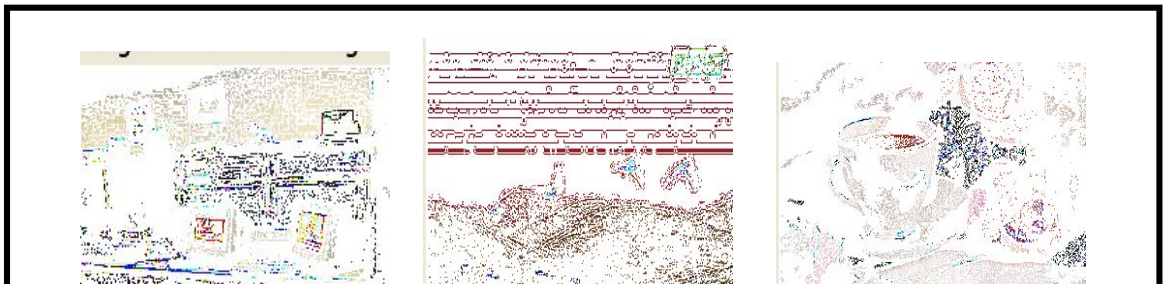


Figure1: (a)The group of cover images ,and (b) the group of Watermarked image.



a) Computer Brides Rose

(b)

Figure 2: (a) The cover images after laplacian sharpening, and (b) The highest values of these images.

```
  .  '  :0.015625  :011010'

     '  :0.015625  :011001'

     'b  :0.0078125  :110111'

     'd  :0.039063  :00100'

     '.  :0.0078125  :110110'

Entropy =4.1494
Average length =4.1953
Redundancy =1.1067
Sequence :
In this work we propose a new algorithm through which we can

embed more data than the regular methods under spatial domain.


Encoded Sequence :
011000111000001001010100111101000000011010000111011000000000011011110
Decoded Sequence :
In this work we propose a new algorithm through which we can

embed more data than the regular methods under spatial domain.
```

Figure 3: The secret data and Huffman coding for it.

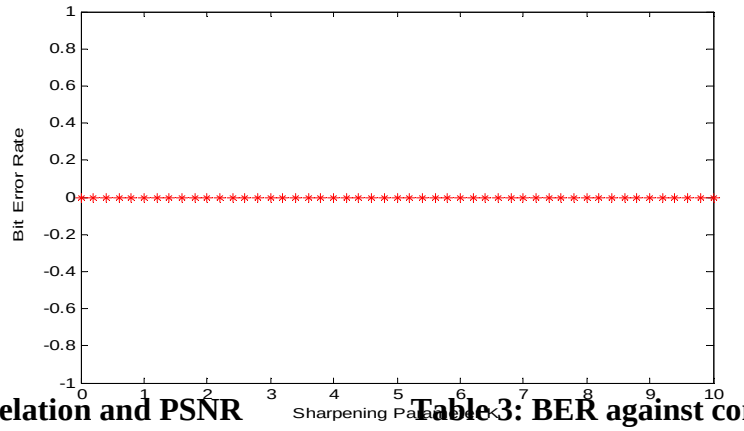
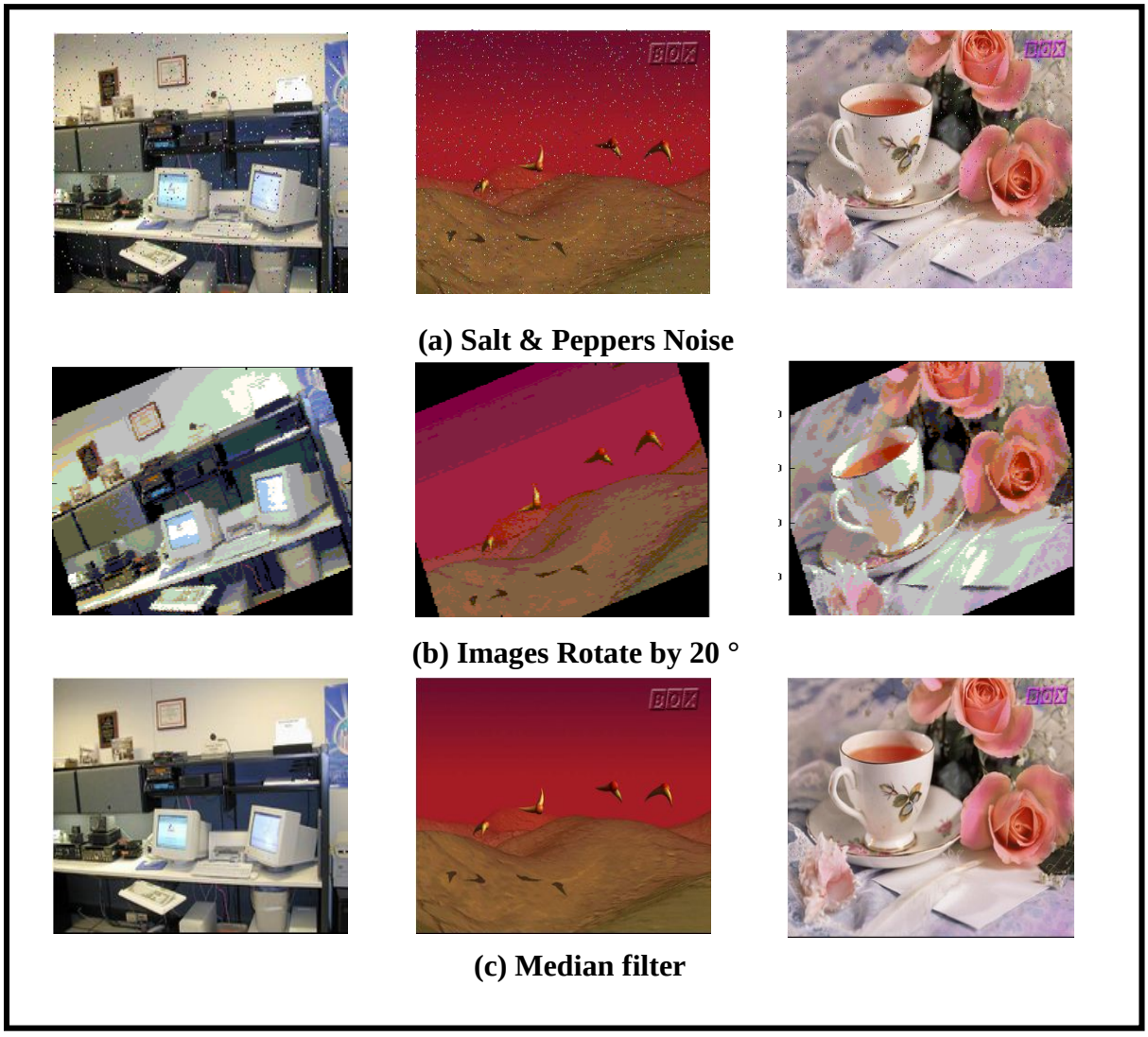


Table2: The Correlation and PSNR values of watermarked test images

Table 3: BER against common attack

Images	PSNR (dB)	Correlation
Brides (320×240)	45.44	0.999334
Computer (230×173)	44.04	0.999503
Rose (320×240)	45.45	0.999402

Attack/Image	Computer	Birds	Rosen
Median filter (3*3)	0.0078	0.0055	0.0042
Salt & peppers Noise	0.25132	0.11719	0.010417
Rotate 20 °	0.1321	0.2533	0.213



(a) Salt & Peppers Noise

(b) Images Rotate by 20 °

(c) Median filter

Figure 5: Watermarked images after attack.

5. Conclusion

1. The combination of the watermarking with the Huffman coding techniques is used in this work to increase the level of security. Even if the attacker knows the embedded text, it is difficult for him to known the original, since it is scrambled before embedded.

2. According to Figure (1), the embedding distortion is very small, and it can't seen by human eyes.

3. According to Table (1) the results that are obtained from the two objective tests prove that the system is secure since the correlation value approaches one and high PSNR value (PSNR equals up to 45.00dB).

4. The evaluation of the proposed method shows good performance as far as invisibility and robustness is concerned. The proposed scheme behaves well in various common signal processing methods as filtering, noise and rotating**.**

5. The selection of the cover image is very important to improve the system performances.

References

[1]H.Kamran, M. Adeel , and S.A.M. Gilani," Digital Image Watermarking in the Wavelet Transform Domain", 2006.

[2] D. Lalitha Bhaskari, P. S. Avadhani, M. Viswanath2, "A Layered Approach for Watermarking In Images Based On Huffman Coding", International Journal on Computer Science and Engineering, Vol. 02, No. 02, PP. 149-154, 2010.

[3] X. Jun and W. Ying, "A Semi-fragile Watermarking Tolerant of Laplacian sharpening", International Conference on Computer Science and Software Engineering, 2008.

[4] H.Kathryn, "Hiding Behind Corners: Using Edges in Images for Better Steganography", proceedings of the second computing women congress, 2006.

[5] http://www.cs.duke.edu/csed/poop/huff/info/#forest.

[6] N. John. and K. Panagiotis," A Wavelet-Based Watermarking Method Exploiting the Contrast Sensitivity Function",International Journal of Signal Processing , 2007.

[7] K. Sushi, and S.k.Muttoo, An overview of Wavelet-like Transform and Image Data Hiding", Proceeding of the 4th National Conference INDIA, 2010.

العلامة المائية للصورة اعتمادا على ترميز هوفمان ومرشح لابلاس

منى مجيد لفتة

جامعة بغداد / كلية التربية للبنات / قسم علوم الحاسبات

الخلاصة

في هذا البحث نقدم خوارزمية نستطيع من خلالها تضمين بيانات أكثر من الأساليب العادية في إطار المجال المكاني . حيث قمنا بضغط البيانات المراد اخفائها باستخدام ترميز هوفمان ومن ثم تضمين البيانات المضغوطة باستخدام طريقة لابلاس للعلامة المائية.

وقد استخدمنا مرشح لابلاس للصورة لتحديد أماكن اختباء فعالة ، ثم على أساس قيمة العتبة وجدنا الأماكن التي توجد بها أعلى القيم الناتجة من هذا المرشح واستعمالها كمواقع لاخفاء العلامة المائية. هدف البحث هو زيادة سعة المعلومات المضمنة من خلال الترميز بطريقة هوفمان وفي الوقت نفسه زيادة الامنية من خلال اخفاء البيانات في الاماكن ذات القيم العليا للحافات والتي تكون اقل ملاحظة من غيرها.

اداء الخوارزمية تم تقييمه من خلال تقنيات الكشف مثل نسبة قمة الاشارة للضوضاء لمعرفة مدى التشوه ،معادلة التشابه الذاتي بين الصور الاصلية والصور بعد الاخفاء و معدل الخطأ لمعرفة مدى قوة الخوارزمية المقترحة ، تطبيق بعض طرق الهجوم منها مرشح لابلاس ، اضافة الضوضاء، المرشح الوسطي، والتدوير. وتبين النتائج انه لا يوجد فرق ملاحظ بين الصور الاصلية والصور بعد الاخفاء سواء من النظر او من نسبة قمة الاشارة للضوضاء، كما نرى ان الخوارزمية المقترحة يمكن أن تقاوم الهجوم بمرشح لابلاس مع اي قيمة للمعامل ،كذلك تظهر قوة الخوارزمية وبشكل جيد لبعض انواع الهجوم مثل التدوير، الضوضاء، والمرشحات.