Using of JPEG image as a cover media for text hiding

Naseef Husam Mohammad
Ahmed Abdul-latief Mohammed
Msc in Computer Science Electronic computer Center
Almustansiriyah University, Baghdad, Iraq

Abstract

This paper, will overview the use of data hiding techniques in JPEG digital image file format. In particular, will describe how one can use Steganography to hide information in a JPEG digital image. This research will review of hiding data by using Least Significant Bit (LSB) as a method, and hiding a text (document) in a JPEG image as a media cover. This paper was implemented by using matlab to complete hiding data. Data hiding is a technique that is used to embed secret information into a cover media. However, the transmitted images may be compressed or not. When it is transmitted, an errors transmitting may occur. If such errors occurred, the receiver cannot extract the information correctly from the cover_image. Digital media have been massively produced, easily manipulated, and swiftly transmitted to almost anywhere in the world at anytime.

1- INTRODUCTION

The steganography is now become an important field of investigation in image processing. Its main application is the watermarking which aims at ensuring image security or authentication. Several methods have been developed under such assumptions as, for instance, spread-spectrum [1, 2]. Another use of steganography is the data-hiding. It is not dedicated to security or authentication, but rather to the embedding of a huge quantity of data in images. Those data can be totally independent from the image content as well as they can enrich it. Important features are here the embedding capacity and the transparency. The capacity is related to the length of the hidden message and the transparency is the ability of data to be invisible to human perceptions. However, data-hiding methods have to be robust to usual transformations such as the compression. The compression standard for images is currently the JPEG one [3]. Several data-hiding methods allow to be robust against the JPEG compression. In particular, the simplest ones proceed similarly as JPEG and then hide data in the frequency coefficients by substitution of their less significant bit

(LSB) [4, 5, 6]. In general, the histogram shifting technique has achieved dramatically improved performance in terms of embedding capacity versus visual quality of stego_image measured by PSNR (Peak Signal to Noise Ratio). A modification to each pixel's LSB produces variations in file. If the embedded file's histogram can identify the existence of the hidden message.

2- JPEG Image

JPEG is the short for Joint Photographic Experts Group. JPEG is a lossy compression technique for color images. Although it can reduce files sizes to about 5% of their normal size. Because of lossy nature of JPEG compression technique some of modifications will appear on image pixels, therefore some detail is lost in the compression. JPEG is "lossy," meaning that the decompressed images isn't quite the same as the one you started with. JPEG images can be viewed in an image program like Microsoft Paint, in web browsers and JPEG viewers. [7]

JPEG is designed for compressing color or gray-scale images of natural, real-world scenes. JPEG handles only still images. JPEG is designed to exploit known limitations of the human eye, notably the fact that small color changes are perceived less accurately than small changes in brightness. JPEG is designed to exploit known limitations of the human eye, notably the fact that small color changes are perceived less accurately than small changes in brightness. Thus, JPEG is intended for compressing images that will be looked at by humans. There are two good reasons: to make your image files smaller, and to store 24-bit-per-pixel color data instead of 8-bit-per-pixel data. Making image files smaller is a win for transmitting files across networks and for archiving libraries of images. The second fundamental advantage of JPEG is that it stores full color information: 24 bits/pixel (16 million colors). The real disadvantage of lossy compression is that if you repeatedly compress and decompress an image, you lose a little more quality each time. [8]

3- STEGANOGRAPHY

The definition of Steganography is literally, covered writing [9]. The idea behind Steganography is to pass a hidden message within another seemingly harmless message so that no one determines that hidden communication is taking place. Cryptography, on the other hand, relies on the cipher to protect the message from being decoded regardless if the message is detected and intercepted. Different types of Steganography have been used throughout the ages. One of the first documented uses of steganography was back in ancient Greece. Text was commonly written on tablets covered with wax. Demeratus wanted to warn Sparta on an imminent invasion from Xerxes. In order to hide the message he scraped wax off a table and wrote a message. The table was then covered with wax again so that the sentries would not notice anything unusual during inspection [10]. Today, with new technologies and the abundance of processing power available at relatively cheap cost, there have been many tools developed to hide messages in email, embed data in images, and pass information through video and sound [5].

4- USES OF STEGANOGRAPHY

Steganography has a wide area of uses. For example, it can be used for hiding data in digital images, e-commerce, and the transport of sensitive data. This is useful to prevent others from reading or knowing our sensitive data. Also it is known as a kind of encryption [1].

5- STEGANOGRAPHIC METHODS

The following formula provides a very generic description of the pieces of the steganographic process:

cover_medium + hidden_data = stego_medium

In this formula, the cover_medium is the file in which the hidden_data will be embedded. The resultant file is the stego_medium (which will be of the same type of file as the cover_medium). The cover_medium (and, thus, the stego_medium) are typically image or audio files. This article will focused on color image files and will, therefore, refer to the cover_image and stego_image [4,6].

6- Least –Significant Bit (LSB) Encoding

Current methods for the embedding of messages into image covers fall into three categories: Least-Significant Bit embedding (or simple embedding), transform techniques, and methods that employ perceptual masking [4].

Least Significant Bit (LSB) insertion is called the simplest approach to hiding data within an image file. In this method, the binary representation took of the hidden_data and overwrite the LSB of each byte within the cover_image. A digital image consists of a matrix of color intensity values. In a typical gray scale image, 8 bits/pixel are used. In a typical color image, there are 24 bits/pixel, 8 bits assigned to each color components.

In general, steganographic techniques embed the bits of the message directly into the LSB plane of the cover_image in a deterministic sequence.

For example, suppose that an image pixel values have the following binary representation as shown in table (1) bellow:

Table (1): The original pixel values and their binary representation

Pixel Values	202	156	91	139
Binary Representation	110010 $\boxed{10}$	100111 $\boxed{00}$	010110 $\boxed{11}$	100010 $\boxed{11}$

(LSB of each byte is shown inside small boxes)

Let's assume that, the value 147 (10010011) needed to be embedded in the pixels shown in table (1). In this case, the LSB of each binary representation of pixel value is replaced the corresponding bits from the value (that is 147 in binary 10010011) we want to embed or hide. So the values will be changed as shown in table (2):

Table (2): Embedding (Hiding) data in the image pixels

Resulted Pixel Values	202	157	88	139
Binary Representation	110010 $\boxed{10}$	100111 $\boxed{01}$	010110 $\boxed{00}$	100010 $\boxed{11}$

Modulating the LSB does not result in a human-perceptible difference because the amplitude of the change is small. Other techniques “process” the message with a pseudorandom noise sequence before or during insertion into the cover_image. The advantage of LSB embedding is its simplicity and many techniques use these methods. LAB embedding also allows high perceptual transparency.

7- STEGANALYSIS

Steganography is the art of invisible communication. Its purpose is to hide the very presence of communication by embedding messages into innocuous-looking cover objects. As long as an electronic document contains perceptually irrelevant or redundant information, it can be used as a cover for hiding secret messages. Each steganographic communication system consists of an embedding algorithm and an extraction algorithm. To accommodate a secret message, the original image, also called the cover_image, is slightly modified by the embedding algorithm. As in cryptanalysis, we assume that the steganographic method is publicly known with the exception of a secret key. The method is secure if the cover_images should have the same statistical properties as the set of cover_images. If there exists an algorithm that can guess whether or not a given image contains a secret message with a success rate better than random guessing, the steganographic system is considered broken [11–13].

The ability to detect secret messages in images is related to the message length. Each steganographic method has an upper bound on the maximal safe message length (or the bit-rate expressed in bits per pixel or sample) that tells us how many bits can be safely embedded in a given image without introducing any statistically detectable artifacts.

Determining this maximal safe bit-rate (or steganographic capacity) is a nontrivial task even for the simplest methods [14]. Recently, a more stringent estimate has been derived using dual statistics steganalysis. The choice of cover_image is important because it significantly influences the design of the stego system and its security. Images with a low number of colors, computer art, images with a unique semantic content, such as fonts, should be avoided. [15,16]. Grayscale images recommended as the best cover_images. The choice of the image format also makes a very big impact on the design of a secure steganographic system. [17]. Indeed, some researchers do not consider those formats for steganography claiming that exchanging uncompressed images is "equivalent" to using cryptography [18].Recently, the JPEG format attracted the attention of researchers as the main steganographic format due to the following reasons: It is the most common format for storing images, JPEG image are very abundant on the internet bulletin boards and public internet sites, and they are almost solely used for storing natural images [4].

8- FIDELITY CRITERIA

The loss information, often associated with stego_image may be acceptable or not depending on predefined tolerated level. Fidelity measures are often used to measure the amount of information losses produced by performing certain hiding algorithm [19].

Fidelity Criteria can be divided into two classes:

Objective and Subjective fidelity criteria .The objective criteria is used for evaluation the stego_image compared with the original (cover_image), a commonly used performance measure applied in this field is the Peak Signal to Noise Ratio(PSNR). The mathematical formula is:-

$$PSNR=10\log_{10}\frac{255^2}{MSE} \qquad \text{…………………………… (1)}$$

And the Mean-Square Error (MSE) is defined as:-

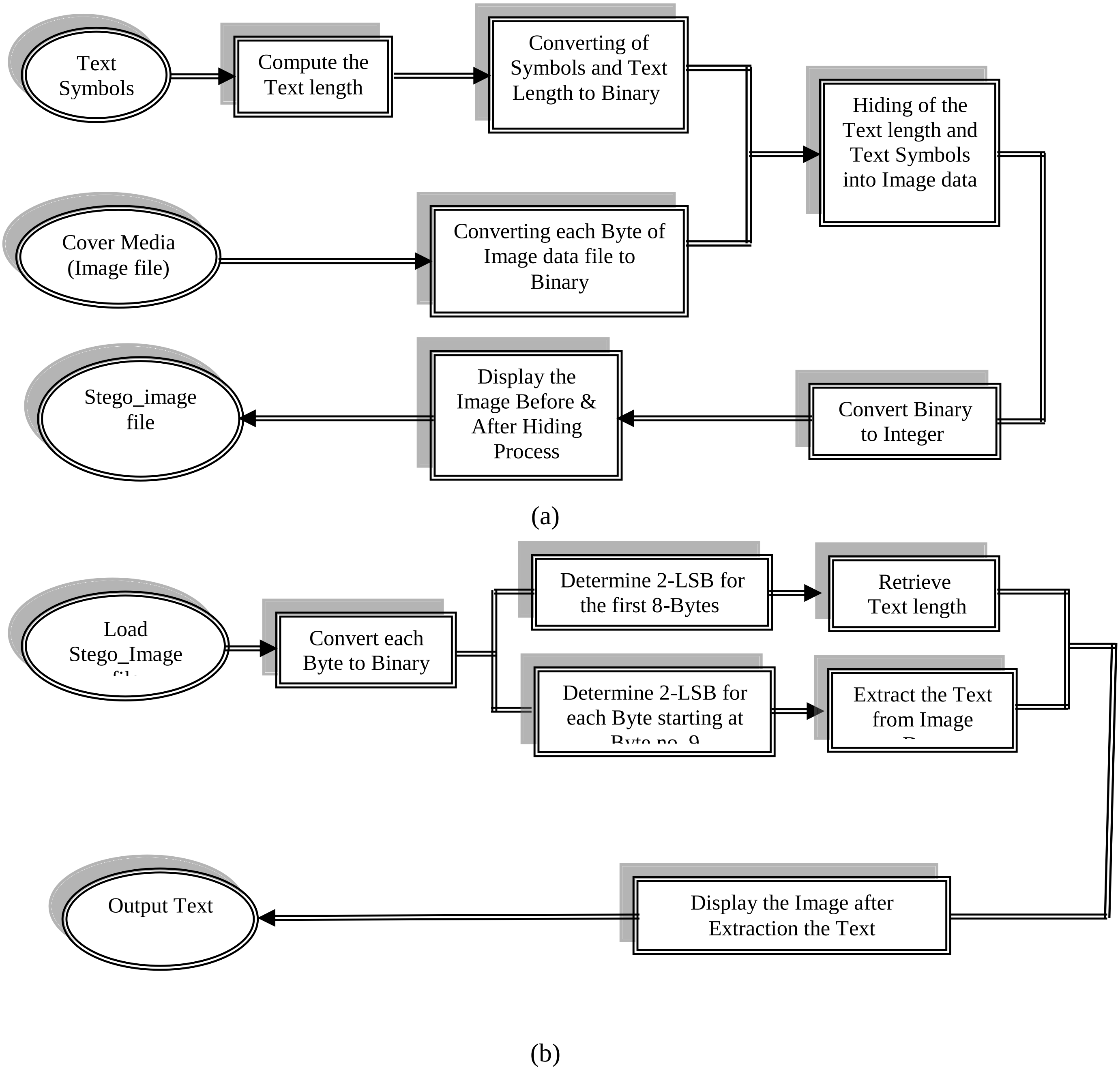
$$MSE=\left(\frac{1}{H*W}\right)\sum_{i}^{H}\sum_{j}^{W}(x_{ij}-x'_{ij}) \qquad \text{…………………..……….. (2)}$$

Where H and W are the size of the cover_image, x_{ij}: is the original cover_image, and x'_{ij} : is the Stego_image [20] .

For a complete evaluation of the stego_image, the objective criteria should be supplemented by subjective criteria. The subjective method often depends on visual inspection.

9- THE PROPOSED SYSTEM

Figure (1) shows the pictorial representation of the proposed system for hiding and re-extracting of the text into the image (cover_media).


Text Symbols
Compute the Text length
Converting of Symbols and Text Length to Binary
Hiding of the Text length and Text Symbols into Image data
Cover Media (Image file)
Converting each Byte of Image data file to Binary
Stego_image file
Display the Image Before & After Hiding Process
Convert Binary to Integer
Load Stego_Image
Convert each Byte to Binary
Determine 2-LSB for the first 8-Bytes
Retrieve Text length
Determine 2-LSB for each Byte starting at
Extract the Text from Image
Output Text
Display the Image after Extraction the Text

(a)

(b)

Figure (1) Pictorial Representation of the Proposed System (a) Hiding Process (b) Extracting the Hidden Text Process.

9.1- Text Hiding Algorithm

In this work, the text symbols were embedded in JPEG image data bytes using 2-LSB. Each symbol is represented by 1 Byte (8-bits) and it is distributed in 4-image bytes. This algorithm is working by determining the text length, and stores it in 2-bytes. These 2-bytes (16-bits) are hidden in the first 8-bytes of the image data .The text information will be hidden in the image pixel data started at ninth bytes. In this algorithm, the text length and text information are embedded in the first byte(Red) of each image pixel, if the last pixel is arrived, the algorithm will uses the second byte(Green) and then the third byte(Blue) of each image pixel until the whole text is hided. The algorithm steps are:-

1. Entering text.
2. Determining the length of the text.
3. Converting of each text symbols into binary system.
4. Loading of image file, each image pixel is stored in 3-Bytes, and converting each of these bytes into binary representation.
5. Converting of text length into binary system (16-bits).
6. Hiding the text length bits into first 8-Bytes of image data bytes.
7. Hiding of text symbols into image data bytes started at the byte number nine.
8. Displaying the image before and after hiding process.

9.2- Re-extraction algorithm

After the text symbols have been hidden as in section 9.1, the algorithm must be completed with the other part of the process (extraction algorithm).

In this section, the scheme of the proposed text re-extraction algorithm will be presented.

The following steps will be addressed to re-extract the text symbols:-

1. Load cover_image file.
2. Convert each byte to binary representation.
3. Determining the 2-LSB for first 8-Bytes, and then compute text length from these bytes.
4. Determining the 2-LSB for each byte started at byte number nine.
5. Arranging each 8-bit extracted from step 4 above, and convert each of these 8-bits (1 Byte) to a symbol.
6. Display the image after text has been retrieved.
7. Display the text symbols.

10- RESULTS

In this work, the proposed system has been tested by using some texts as hidden_data embedded in some JPEG images as cover_media .Three texts different in length and three images different in complexity are presented in table(3) :

Table (3): The texts and images used in the system

Texts & Images	Length & Complexity Description
Text 1	Short length text
Text 2	Medium length text
Text 3	Long text
Image 1	Simple image
Image 2	Medium complexity image
Image 3	Complex image

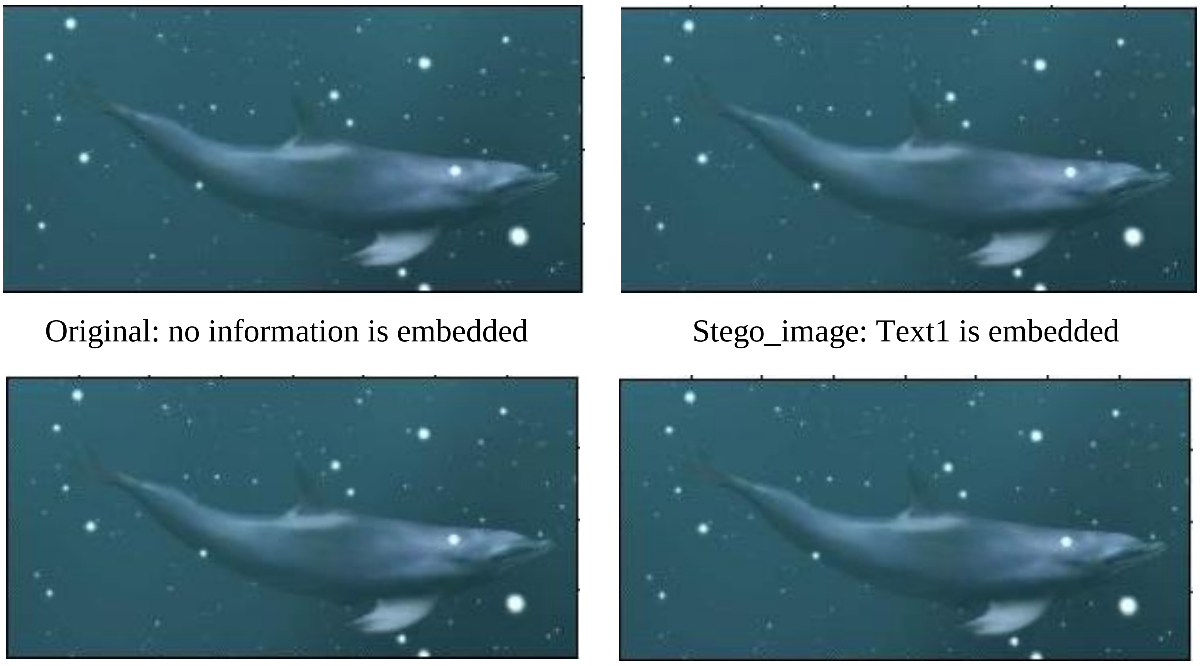
The estimated results using equations (1) and (2) are introduced in table (4) and these results will show the efficiency of the proposed system.

Table (4): The results obtained by hiding 3 texts in 3 images

Images	Texts	PSNR
Image1	Text 1	45.6614
	Text 2	43.0909
	Text 3	39.5091
Image2	Text 1	44.9223
	Text 2	44.2874
	Text 3	41.5110
Image3	Text 1	43.6545
	Text 2	42.7492
	Text 3	40.4389

From table(4), one can find that using simple, medium, or complex image, and hiding short, medium, and long texts, the PSNR is not widely differs. So embedding text in an image can be used without any effect in stego_media especially if the stego_media is a JPEG image type .

Figure (2) demonistrates the resulted stego_images when different length texts were embedded.



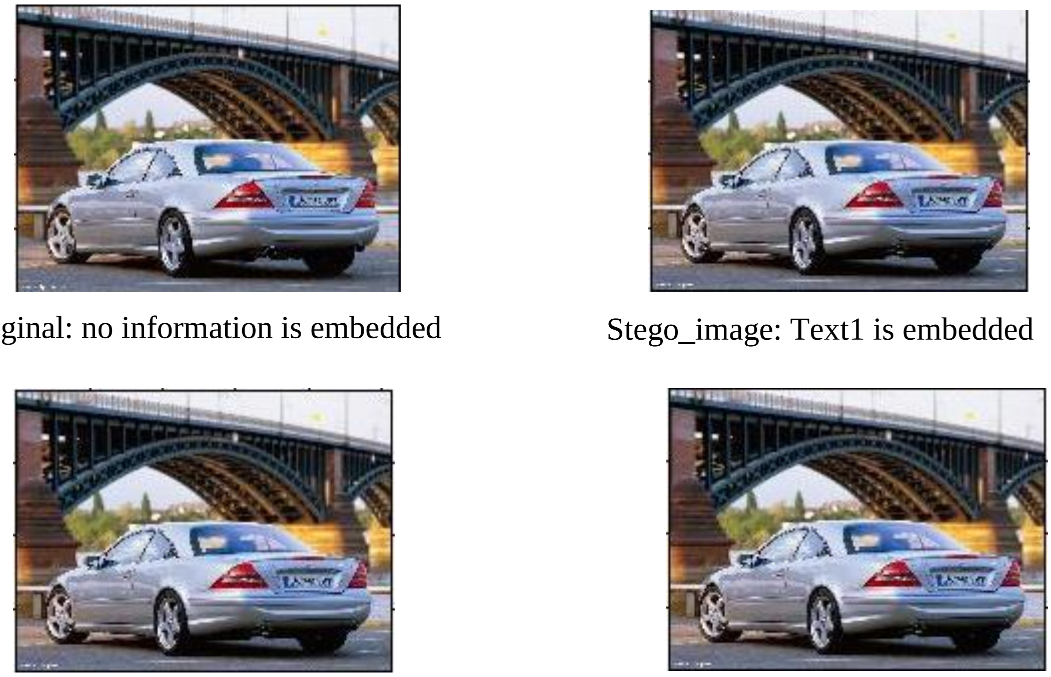
Original: no information is embedded

Stego_image: Text1 is embedded

Stego_image: Text2 is embedded

Stego_image: Text3 is embedded

(a)



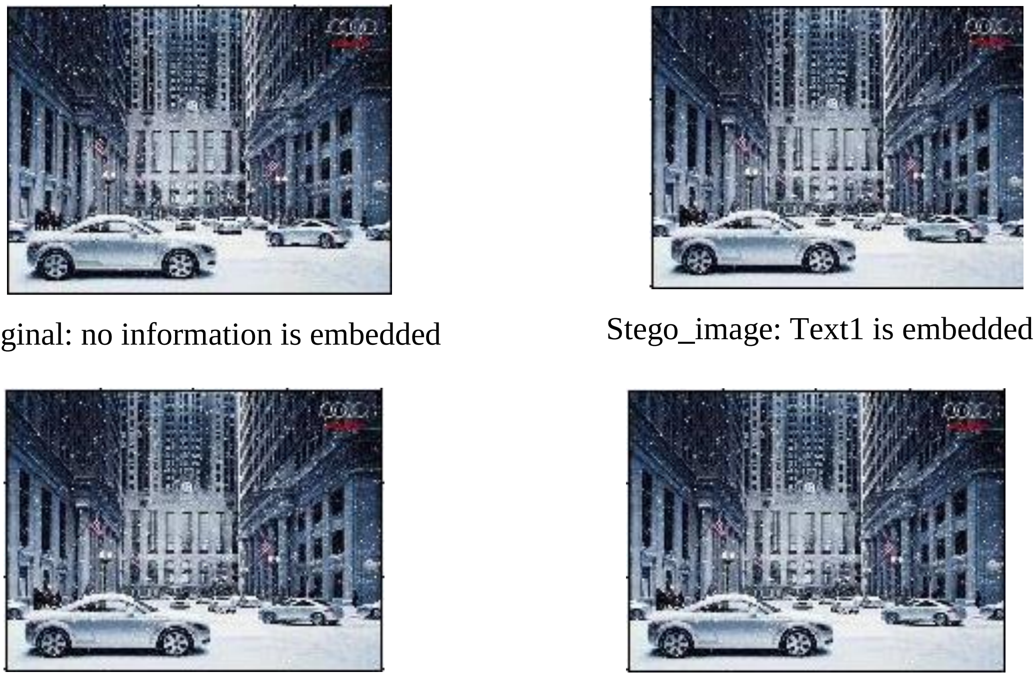
Original: no information is embedded

Stego_image: Text1 is embedded

Stego_image: Text2 is embedded

Stego_image: Text3 is embedded

(b)



Original: no information is embedded

Stego_image: Text1 is embedded

Stego_image: Text2 is embedded

Stego_image: Text3 is embedded

(c)

Figure (2): Results of presented information hiding process: a) The three texts are hidden in image1, b) The three texts are hidden in image2, and c) The three texts are hidden in image3.

However, from the figure (2) above, the results show that all images (original, and stego images conyaining the three different length texts) are virtually indistinguishable, with excellent PSNRs. To the regard of PSNR, the higher it is, the closer both images are.

11- CONCLUSION

In this work, a software package was presented which can be used for text words hiding and re-extracting. On a small number of test subjects shows that steganography effectively encodes hidden messages in media files without the viewer being able to notice–even if they have the original cover file to compare to it. Results obtained by attempting to embed different text lengths explained that long text length may not cause big effects on cover_medium.

This research concludes that steganography is effective for hiding messages without altering the cover file noticeably. The goal of achieving protection of large amounts of embedded data against intentional attempts at removal may be unobtainable.

12- References

1. J. Ingemar Cox, Joe Kilian, T. Leighton, and Talal Shamoon. Secure spread spectrum watermarking for images, audio and video. In Proccedings IEEE ICIP, volume 3, pages 243–246, 1996.
2. Brian Chen and Gregory W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Transactions on Information Theory, 47(4):1423–1443, 2001.
3. G. Wallace. "The JPEG still picture compression standard". Communication of the ACM, 34(4):31–44, Apr. 1991.
4. C.-C. Chang, T.-S. Chen and L.-Z. Chung. "A Steganographic Method Based upon JPEG and Quantization Table Modification". In Information Sciences, volume 141, pages 123–138, 2002.
5. C.-T. Hsu and J.-L. Wu. "Hidden Digital Watermarks in Images. In IEEE Transaction on Image Processing", volume 8, pages 58–68, 1999.
6. H.-W. Tseng and C.-C. Chang."High Capacity Data Hiding in JPEG-Compressed Images". Informatica, 15(1):127–142, 2004.
7. http://education.qld.gov.au/about/files/jpg.html.
8. http://www.faqs.org/faqs/jpeg-faq/.
9. Dr Anthony Ho, Internet Business 99, Singapore, 24th June 1998, Page 2, http://www.datamarktech. com/publications/steganography.pdf
10. F. Niel Johnson, Steganography, http://www.jtc.com/stegdoc/stegdoc.html, Section 2.2.
11. R.J. Anderson and Petitcolas, F.A.P.: On the Limits of Steganography. IEEE Journal of Selected Areas in Communications: Special Issue on Copyright and Privacy Protection), Vol. 16(4) 474−481 (1998).
12. C. Cachin :"An Information-Theoretic Model for Steganography". In: Aucsmith D. (eds.): Information Hiding: 2nd International Workshop. Lecture Notes in Computer Science, Vol. 1525. Springer-Verlag, Berlin Heidelberg New York 306–318 (1998).
13. S. Katzenbeisser and F.A.P. Petitcolas :"On Defining Security in Steganographic Systems". Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, Vol. 4675. San Jose, California (2002).
14. R. Chandramouli and N. Memon :"Analysis of LSB Based Image Steganography Techniques". Proceedings of ICIP 2001 (CD version). Thessaloniki, Greece (2001).
15. J. Fridrich , M. Goljan , and R. Du :"Reliable Detection of LSB Steganography in Grayscale and Color Images". Proc. of ACM: Special Session on Multimedia Security and Watermarking. Ottawa, Canada 27–30 (2001).
16. J. Fridrich, M. Goljan, and R. Du : "Detecting LSB Steganography in Color and Grayscale Images". Magazine of IEEE Multimedia: Special Issue on Security, Vol. Oct-Dec 22–28 (2001).

17. T. Aura : "Practical Invisibility in Digital Communication". In: Anderson, R.J. (eds.): Information Hiding: 1st International Workshop. Lecture Notes in Computer Science, Vol.1174. Springer-Verlag, Berlin Heidelberg New York 265−278 (1996).

18. J.J. Eggers, R. Bäuml, and B. Girod :"A Communications Approach to Image Steganography". Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents, Vol. 4675. San Jose, California (2002)

19. E. Scott umbaugh: "Computer Vision and Image Processing: A practical Approach Using CVIP Tools", Prentice Hall, 1998.

20. Aiad Ibraheem Abdul-sada: "Hiding Data Using LSB-3", J.Basrah Researches (sciences) vol.33. No.4.(81.88)DEC. (2007)

استخدام الصور الرقمية كوسط لأخفاء النصوص

الخلاصة

يقوم هذا العمل بعرض كيفية إستخدام تقنيات إخفاء البيانات في ملفات الصور الرقمية من نوع JPEG . بشكل خاص، سوف يتم اعطاء وصف لكيفية إستخدام تقنيات الاخفاء لإخفاء المعلومات في الصور الرقمية من نوع JPEG . يستعرض هذا البحث إخفاء البيانات بإستخدام طريقة (LSB) وكيفية استخدام صورة من نوع JPEG كوسط لإخفاء المعلومات. وقد تم تنفيذ هذا العمل باستخدام برنامج الماتلاب. إخفاء البيانات عبارة عن تقنية تستخدم لتضمين المعلومات السرية في وسط الإخفاء.

إن الصور التي يتم ارسالها عبر شبكة الحاسبات قد تكون مضغوطة او غير مضغوطة. إن عملية الإرسال قد يصاحبها بعض الأخطاء. عند حدوث مثل هذه اللأخطاء فإنه لايمكن لمستلم البيانات إسترجاع المعلومات بشكل صحيح من وسط الاخفاء (الصورة الرقمية). توجد أنواع كثيرة من وسائط الاخفاء الرقمية، وهذه الوسائط يمكن معالجتها بسهولة، وكذلك يمكن إرسالها بكل سهولة ويسر الى أي مكان في العالم وفي أي وقت.