Digital Watermarking Technique for Hiding Text Into Image[1]

Ismael A. Jannoud [2] **Mohammed A. F. Al-Husainy[3]**

Abstract

The recent growth of networked multimedia systems has increased the need for the protection of digital media. This is particularly important for the protection and enforcement of intellectual property rights. Digital media includes text, digital audio, images, video and software. Many approaches are available for protecting digital data; these include encryption, authentication and time stamping. In the computer world, watermarking techniques can be used to hide secret messages in digital images, movies or sound. In this paper, a technique for hiding a message into digital images (256 gray levels) is proposed. Some tests, on the proposed technique, are done by applying the technique in hiding messages into some images. From the recorded results, we can note that the proposed watermarking technique introduces, in many cases, a reasonable way that can be used to hide secret messages in the digital images.

[1] For the paper in Arabic see pages (41-42).

[2] Department of Electronics and Communications, FMEE, Damascus University.

[3] Department of Computer Science, Faculty of Sciences and IT, Al-Zaytoonah University of Jordan.

1. Introduction:

A watermark is a secret code or image incorporated into an original image. The use of perceptually invisible watermarks is one form of image authentication. A watermarking algorithm consists of three parts: the watermark, the marking algorithm and the verification algorithm. Each owner has a unique watermark. The marking algorithm incorporates the watermark into the image. The verification algorithm authenticates the image, determining both the owner and the integrity of the image [1].

Digital watermarking helps owners in asserting their intellectual property rights on the artistic works. Figure 1 shows that the basic components of any watermarking technique consist of a marking algorithm that inserts information, the watermark (message), into an image. The watermark is inserted into the image in the spatial domain or spatial frequency domain. As part of the watermarking technique, a testing algorithm must be defined that tests an image to see if a particular watermark is contained in the image [2].

When an image is marked using a fragile watermark, an attacker does *not* want to make changes to the image that alter the mark; it is desired by the attacker to have an altered image "pass" as authentic. Here, an attacker wants to remove the watermark at a minimal loss in image quality. In this way the true owner cannot verify the presence of the watermark in the image, thus greatly reducing any ownership claim. The watermark, therefore, must remain in the image after many types of attacks. These attacks include compression to low data rates, filtering, printing and rescanning, in addition to geometric attacks such as cropping, resampling and rotation. Furthermore, users must not be able to attack the watermark through collusion by comparing multiple copies of the image marked with different watermarks. These watermarks are known as *robust watermarks*[2].

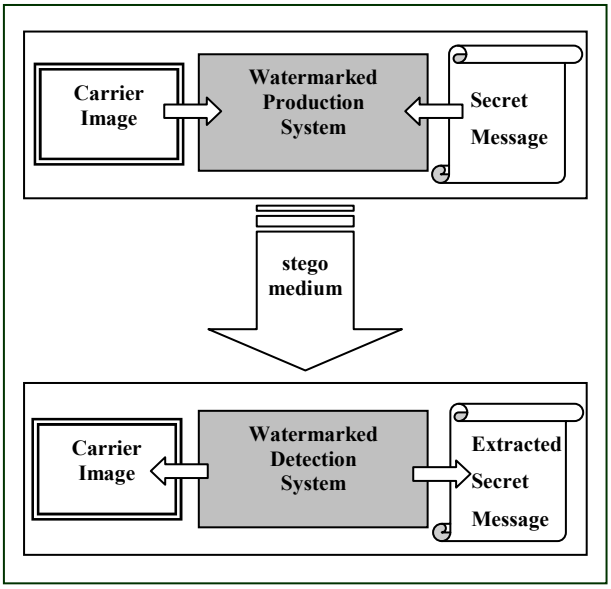

Figure 1: Watermarking System.

In this paper we will describe invisible watermarks that are designed to use perceptual information based on human visual system models. There are three main principles that characterize perceptually based watermarks:

1. **Transparency:** the watermark is not visible in the image under typical viewing conditions.
2. **Robustness against attacks:** the watermark can still be detected after the image has undergone linear or nonlinear operations.
3. **Capacity:** the watermarking technique must be capable of allowing multiple watermarks to be inserted in an image, with each watermark still being independently verifiable.

2. Techniques Of Watermarking:

Information may be covered by coding as in *cryptography* or by hiding as in watermarking (*steganpography)*. In this section, we present an overview of techniques that can be used for hiding the digital data, from an application point of view. Many common digital hiding techniques employ graphical images or audio files as the carrier medium.

There are many ways in which messages can be hidden in digital media. Digital forensics examiners are very familiar with data that remains in file slack or unallocated space as the remnants of previous files and, of course, one can write programs that can access slack and unallocated space directly. Small amounts of data can also be hidden in the unused portion of file headers [3].

Information can also be hidden on a hard drive in a secret partition. A hidden partition will not be seen under normal circumstances although disk configuration and other tools might allow complete access to the hidden partition [4]. This theory has been implemented in a steganographic ext2fs file system for Linux. A hidden file system is particularly interesting because it protects the user from being inextricably tied to certain information on their hard drive. This form of *plausible deniability* allows a user to claim not to be in possession of certain information or to claim that certain events never occurred. Under this system, users can hide the number of files on the drive, guarantee the secrecy of the files' contents, and not disrupt non-hidden files by the removal of the stego file driver [5, 6, 7].

Another digital carrier can be the network protocols themselves. Covert TCP by Craig Rowland, for example, forms covert communications channels using the Identification field in Internet Protocol (IP) packets or the Sequence Number field in Transmission Control Protocol (TCP) segments [4, 8].

Image and audio files remain the easiest and most common carrier media on the Internet today because of the plethora of potential carrier files already in existence, the ability to create an infinite number of new carrier files, and the easy access to stego software that will operate on these carriers. For that reason, we will return our focus back to image and audio files.

The most common stego method in audio and image files employs some type of *least significant bit (LSB) substitution* (or *overwritin*g). The

"least significant bit" term comes from the numeric significance of the bits in a byte. The high-order, or most significant, bit is the one with the highest arithmetic value (i.e., 2^7 =128) while the low-order, or least significant, bit is the one with the lowest arithmetic value (i.e., 2^0=1) [9]. Newer, more complex, steganography methods continue to emerge. *Spread spectrum steganography* methods are analogous to spread spectrum radio transmissions (developed in World War II and commonly used in data communications systems today) where the "energy" of the signal is spread across a wide frequency spectrum rather than focused on a single frequency, in an effort to make detection and jamming of the signal harder. Spread spectrum stego has the same function; avoid detection. These methods take advantage of the fact that little distortions to image and sound files are least detectable in the high energy portions of the carrier; i.e., high intensity in sound files or bright colors in image files. Even when viewed side-by-side, it is easier to fool human senses when small changes are made to loud sounds and/or bright colors [10].

3. The Proposed Technique:

In this paper, images of 256 gray levels are used to hide a text of alphabetic characters (‘a’..’z’) with ‘space character‘. Consider an image ***I (w×h)*** that is used to hide a text ***T*** of ***n*** characters. Where ***w*** and ***h*** are the width and height of the image, ***n*** is the number of characters in the text ***T***. The technique involves two phases **hiding phase** and **extracting phase**:

In the **hiding phase**, the characters of the secret message text ***T*** are hidden in the image ***I***. The hiding phase will produce an image ***O*** that contains inside its bytes the characters of the secret text ***T***. This phase consists of the following steps:

Step (1): Rescale the gray levels of the image ***I*** from 256 to the number of levels that is equal to the nearest number greater than 27 (such that there are 26 alphabetic characters plus space character ‘ ‘). The scaling operation must keep the *Signal to Noise Ratio (SNR)* of the rescaled image suitable (for example SNR ≥ 25.0 db). This step is done to increase the probability of finding the sequence of the alphabetical characters in the secret message text ***T***.

Step (2): Construct a gray table ***G*** of the gray value from the rescaled image ***I***.

Step (3): Map each alphabetic character and space character to one of the gray value in the gray table in (**Step (2)**) of the rescaled image. The mapping step must map the most frequent gray in the image to the most frequent alphabetic character in the English language. This step will produce a mapping table ***M***.
Step (4): For each character ***k*** in the text ***T*** (where ***k***=1..n). Use the mapping table to get the equivalent gray value that matches to this character, then scan the image ***I*** byte by byte from ***1*** to ***w×h*** and try to find the mach byte (gray) ***j*** in the image ***I*** that is equal to the gray value that is mapped (in **Step (3)**) to the character ***k*** in the text ***T***. Store in the byte after the matched byte ***j*** the displacement (i.e., number of bytes (displacement≤255)) between the current matched character ***k*** and the next matched character ***k+1***. Store 0 after the last match character of the text ***T***. Note the gray value mapped to the first character in the text ***T*** is always stored in the first byte of the image ***I***.
In **extracting Phase**, the secret message text ***T*** will be extracted from the watermarked image ***O***. This phase consists of the following steps:
Step (1): Construct a gray table ***G*** of the gray value from the image ***O***. Note this operation must exclude the bytes in the image ***O*** that are used (through the hiding phase) to store the displacement between the two sequenced bytes, that are matched to the gray value mapped to the two sequenced characters in the secret message text ***T***).
Step (2): Map each alphabetic character and space character to one of the gray value in the gray table in (**Step (1)**) of the image ***O***. The mapping step must map the most frequent gray in the image to the most frequent alphabetic character in the English language. This step will produce a mapping table ***M***.
Step (3): Scan the byte of the image, from the first byte, to extract the bytes (gray values) in the image ***O*** that represent the matched bytes to the gray value mapped to the characters of the original text ***T.*** This scan operation use the byte that represents the displacement byte to move from one matched byte (gray) to the next.
Step(4): For each extracted byte in (**Step (3)**), use the mapping table ***M*** to get the characters in the original text ***T*** that is mapped to a certain gray value in the mapping table M.

4. Experimental Results:

The proposed technique was tested to hide some text in some 256 gray levels images. Table (1) lists some experiments. For more explanation, fig. 2 clarifies the hiding and extracting phases of the proposed technique that are applied in the experiments of table (1).

Table (1): Experimental Results.

Experiment#	Text Length (n)	Image (w×h)	SNR (db) after hiding text
1	1000	Lena (512×512)	25.6952
2	104	Lena (512×512)	30.9371
3	1000	Fruit (512×512)	23.8706
4	104	Fruit (512×512)	27.791
5	1000	Sea (512×512)	24.4065
6	104	Sea (512×512)	30.5339

Experiment #	Original Image	Original Image + Secret Message	Original Image + Highlighting Secret Message
1			
2			

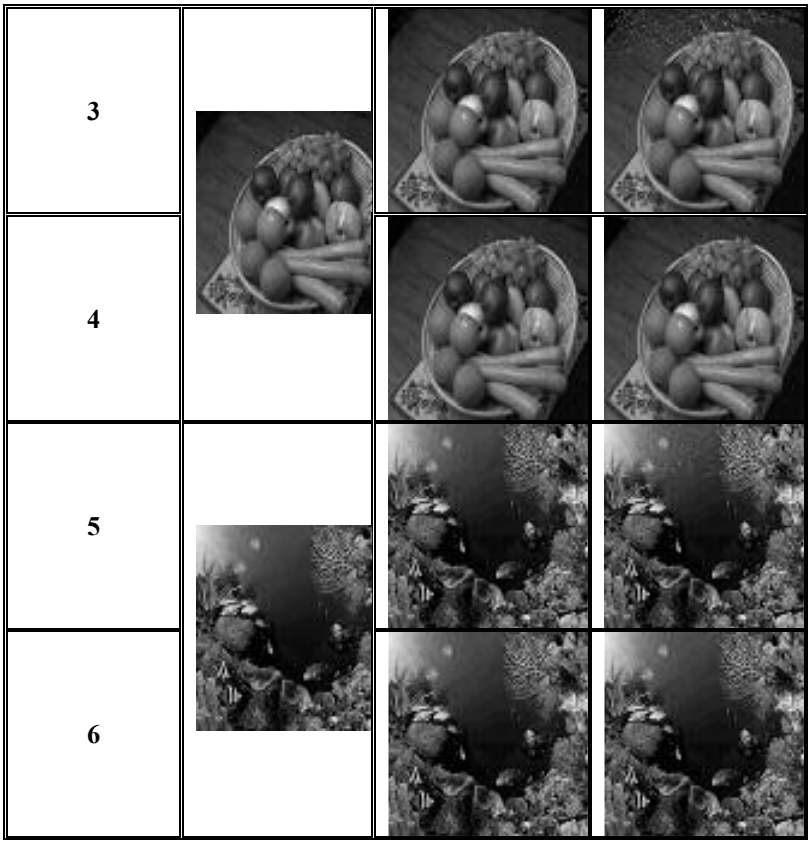

Figure 2: The Hiding And Extracting Phases Of The Experiments.

Conclusion

From the recorded results, we note that the proposed technique can be used to hide messages in the 256 gray levels images in a good way that keeps the original image with a small signal to noise ratio, which means that its hard to detect the hiding message. The efforts in this research will continue to apply this technique on the colored images, and try to enhance the performance and the recorded result in the future.

References

1. Raymond B. Wolfgang and Edward J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies", Video and Image Processing Laboratory (VIPER), School of Electrical and Computer Engineering, Purdue University, Indiana, 47907-1285, USA.

2. Raymond B. Wolfgang, Christine I. Podilchuk, and Edward J. Delp, "Perceptual Watermarks for Digital Images and Video", School of Electrical and Computer Engineering, Purdue University, USA, Email: ace@ecn.purdue.edu.

3. Curran, K. and Bailey, K. "An Evaluation of Image Based Steganography Methods." *Int. J. of Digital Evidenc*e, 2003. URL: http://www.ijde.org/docs/ 03_fall_steganography.pdf. Last accessed: 2003.

4. Johnson, N.F., Duric, Z. and Jajodia, S.G. *Information Hiding: Steganography and Watermarking - Attacks and Countermeasure*s. Norwell (MA): Kluwer Academic Publishers, 2001.

5. Anderson, R., Needham, R., and Shamir, A. "The Steganographic File System." In: Aucsmith, D. (ed.). *Proc. of the Second International Workshop on Information Hiding (IH '98*), Portland, OR, April 1998. *Lecture Notes in Computer Scienc*e, Vol. 1525. New York: Springer-Verlag, 1998.

6. Artz, D. "Digital Steganography: Hiding data within Data." *IEEE Internet Computing*, May/June 2001. URL:http://www.cc.gatech.edu/classes/AY2003/cs6262_fall/ digital_steganography.pdf. Last accessed: 2003.

7. McDonald, A.D. and Kuhn, M.G. "StegFS: A Steganographic File System for inux." In: Pfitzmann, A. (ed.). *Proc. of the Third International Workshop on Information Hiding (IH '99*), Dresden, Germany, Sept.-Oct. 1999. *Lecture Notes in Computer Scienc*e, Vol. 1768. New York: Springer-Verlag, 2000. URL: http://www.cl.cam.ac.uk/~mgk25/ih99-stegfs.pdf. Last accessed: 2003.

8. Rowland, C.H. "Covert Channels in the TCP/IP Protocol Suite." *First Monday*, 1996. URL: http://www.firstmonday.dk/issues/issue2_5/rowland/. Last accessed: 2003.

9. Fridrich, J. and Du, R. "Secure Steganographic Methods for Palette Images." In: *Proc. of The 3rd Information Hiding Worksho*p, September 1999, Dresden, Germany. *Lecture Notes in Computer Scienc*e, Vol. 1768. New York: Springer-Verlag, 2000.

10. Wayner, P. *Disappearing Cryptography - Information Hiding: Steganography & Watermarkin*g, 2nd. ed. San Francisco: Morgan Kaufmann; 2002.

Received, 12 May, 2005.