The Art of Steganography: Watermarking through the Least Significant Bit

Joshua Haskins

230099946

Table of Contents

Introduction

In recent years with the advancement of technology and the swift increase of data transmission, the Internet has become the leading method of data sharing. With the ease of creating and distributing unlimited numbers of copies of data files, intellectual rights have become a major concern [1]. With this has come the need for the enforcement and protection of digital content, as the intellectual property rights still do belong to the creators of digital media pieces. In this new age of digital media, it has required media professionals to innovate and find new methods of embedding copyright information into their digital files. To fill the gap between digital distribution and copyright issues of digital media, digital image watermarking is at the forefront [2]. The techniques used in watermarking are largely based upon steganography.

Digital watermarking allows an individual or organization to add hidden or visible copyright notices and/or other information to different types of digital files, such as audio, video, or even text. This hidden information typically relates to the original owner, such as the authors name or company, date and location. This technique borrows its name from the watermarking of paper materials, such as bank notes (paper money). Though in recent years many countries have migrated to polymer based bank notes, such as Canada, in which the traditional watermark is no more, but additional security features have been added such as raised ink, metallic attributes, and clear windows [3]. A traditional watermark is simply a logo or name embedded into the original image; whereas a digital watermark is a digital pattern or signal embedded into a digital file. With the unmodified duplication or copying of this digital file, the watermark will be present; this serves as the digital signature for the original author.

Methodology

There are many different techniques to hide information inside of images, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Least Significant Bit (LSB) and Discrete Fourier Transform (DFT). In this paper, our primary focus will be on LSB, which is a method of digitally watermarking images. One of the main advantages of LSB is the accuracy and simplicity of the algorithm. Other types of digital watermarking rely upon busy areas of the image, to better hide the adjustments made to the image; or the entire image for watermark insertion. LSB relies on the latter, which is pixel based modifications [4].

> The LSB technique works by replacing some of the information in a given pixel with information from the data in the image. While it is possible to embed data into an image on any bit-plane, LSB embedding is performed on the least significant bit(s). This minimizes the variation in colours that the embedding creates [4].

"The purpose of digital watermarks is to provide copyright protection for intellectual property that's in digital format" [2]. The digital image watermark is the information and/or logo that is to be embedded. The image that this watermark is to be embedded into is called the host image. Perceptibility is an attribute digital watermarking provides. This means that when an image has been watermarked, there should be no visible degradation of the image; this means, no visible artifacts that the image has a watermark and should be indistinguishable from the original image. This indistinguishability should be to the naked human eye and also by using advanced equipment to analyze the image. Robustness is also an important attribute; this means that the watermark is highly resistant to any distortion. This can be through just normal use, such as

cropping the image; or a deliberate attack, such as attempting to alter or remove the watermark. Robustness needs to be able to withstand a wide variety of attacks.

Background

Steganography is the act of hiding data inside of another object. In security, steganography plays a major role [5]. In ancient Egypt, the Egyptians communicated using a hieroglyphic language, which is a language consisting of symbols; by combining these symbols into groups, they were able to communicate covertly. The final result looked similar to a painting, but actually contained a hidden message, only to be understood by those who comprehended the language. Post the Egyptians, the Greeks used steganography, meaning hidden writing, which also is where the name is derived from [5]. "The goal of steganography is to hide the fact that any form of communication is occurring by embedding messages into an innocuous-looking cover medium such as digital image, video, audio and so on, while steganalysis [focuses] on revealing the presence of the secret messages and [extracting] them" [5]. The embedded information can be regular text, encrypted text (adding an additional layer of security), or simply images. Customarily, the steganography tactic is to hide the message in obvious object, so as not to draw attention to it. Primarily, the goal is to prevent the opposition from suspecting the existence of concealed communications, not to thwart the ability to decode the message.

One famous historical example of steganography is that of Histiaeus, who shaved the head of one of his slaves, tattoed a message on his scalp, and then let the hair grow back. Histiaeus then sent his slave to Aristagoras, with a message to shave his head [6]. Thus, the hidden message was delivered. The idea of digital watermarking largely come about in the 1990s

due to the widespread use of the Internet [2]. A similar technology, but for identifying music works was patented by Emil Hembrooke in 1954 [2].

Principles

Steganography entails three different types of obscurity, which are public key steganography, secret key steganography, and pure steganography [5]. Public key steganography involves two keys, a public key and a private key. It involves the sender using the receivers' public key to embed the information. The embedded information can only be detected using the receivers' private key [5]. In this method, the sender is not even able to read the message after it has been embedded using the public key, as he does not have access to the private key. Secret key steganography uses an algorithm that is publically known, and a secret key. This key is selected before any communication occurs, and is known by all of the parties involved. To embed or extract the hidden information, this key is required. If the key were not used, then it would be similar to finding a needle in a haystack. Pure steganography relies upon secret through obscurity; thus, no preceding communication amidst the parties involved is required. The algorithm is not publically known, and breaking this algorithm can be of great pain. If external parties can corroborate the existence of secret information being embedded in an object, then steganography is of no use.

The robustness of watermarks is a very important attribute; the level of robustness will depend on the exact application of use for the watermark. "[Watermarks] can be classified into two types according its function, namely robust watermark for copy- right protection and fragile watermark for integrity verification" [7]. Fragile watermarks fail to be recovered even after the slightest modification, whereas robust watermarks will resist a range of transformations. There is

also a class in between these two, which is called semi-fragile, which will fail after malignant transformations, but resist benign transformations. Another important trait of watermarks is perceptibility. There are three different types of perceptibility: perceptible, imperceptible, and perceptual. Perceptible is where the watermark is visible in the image. A common example of this is the network logo for the TV station in the bottom left or right corner of TV programs. Imperceptible watermarks are undistinguishable from the source object to the watermarked object. Perceptual however is where the object is watermarked in such a way that the human eye cannot detect it, but the watermark is still detectable and recoverable.

Process

When the encoder inserts the watermark into the image, the process of watermarking begins. The image is then sent to the receiver, and the receiver uses a decoder to extract and validate the watermark. A decoder is not required if the watermark is visible in the image. The decoder uses a threshold value to compare the extracted watermark to the intended watermark. If it is below the threshold value, then the watermark is confirmed [2]. If it does not meet the threshold, then the image either has not been watermarked or has been tampered with. This process is similar as to how one would verify physical objects with watermarks, such as bank notes.

Technique Used

As the most common form of watermarking is through LSB [2], I opted to explore this algorithm. LSB works by adjusting the image pixel value in binary form of the furthest right bit (the least significant bit). Lets say we have the pixel value of 10001000 (this is a binary number)

and we need to add a 1 to it, we would end up with the value 10001001. By adjusting the least significant bit, this results in the least amount of colour distortion to the image. Now the actual hidden message to be embedded is of course going to be much, much larger than simply just a 1.

Pixel 1 = 10001000	Pixel 5 = 10101000
Pixel 2 = 10111010	Pixel 6 = 11001000
Pixel 3 = 10010110	Pixel 7 = 10011010
Pixel 4 = 10001000	Pixel 8 = 11110100

Fig. 1. Pixel values

Now lets say we wanted to embed the letter 'J' in the pixel values of Figure 1. First, we need to convert 'J' to binary, which is 01001010. We then modify the least significant bit in each of the eight pixel values. In the end, we are left with Figure 2, where the changed values have been bolded. Now an image is more complex than these simple eight pixels, but this gives the general idea of how LSB works.

Pixel 1 = 10001000	Pixel 5 = 10101001
Pixel 2 = 10111011	Pixel 6 = 11001000
Pixel 3 = 10010110	Pixel 7 = 10011011
Pixel 4 = 10001000	Pixel 8 = 11110100

Fig. 2. Adjusted pixel values

Even though LSB is a very simple method of watermarking, it does have weaknesses. Once the attacker has the algorithm, and the type of LSB used, they would easily be able to read and/or modify the hidden information without either the sender or receiver knowing. But LSB does have some advantages, it survives added noise, lossy compression (meaning parts of the object are lost in compression) and transformations, such as cropping [2]. A method of

completely defeating LSB is to set the bits of the least significant bits of all pixels to either a one or zero, thus wiping out the hidden message completely.

To compare image compression quality, two different error-checking methods can be used [4], [8]. The first method being Peak Signal to Noise Ratio (PSNR), which represents the peak error between the original and modified image. Whereas the second, called Mean Square Error (MSE), calculates the cumulative squared error between the original and modified image. The higher the PSNR value, the better the quality of the modified image, whereas the lower the MSE value the better the quality of the modified image [8]. Both of these methods are used to denote a value of quality between the original and the modified image.

Related Work

A. E. Mustafa et al [5], purpose a new algorithm for LSB based steganography. In this method, the authors purpose that instead of modifying the first LSB (LSB-1), to instead modify the second LSB (LSB-2) and modify the LSB-1 based upon the new value to LSB-2 [5]. The method is done to minimize the differentiation between the two images, having a low MSE and a high PSNR. The algorithm of this method is still quite simple and allows for less distortion due to the imbedded message.

Abdullah Bamatraf, Rosziati Ibrahim and Mohd. Najib Mohd. Salleh [9] and Amit Singh, Susheel Jain, Anurag Jain [10] both present a method of embedding the watermark by first inversing the watermark bit values, then by embedding it into the host image by using a different order than the regular LSB method. The method of embedding is based upon the pixel value at

the embedded location. If the original value is even, then the algorithm will subtract 1, whereas if odd, it will add one to the pixel value.

Shilpa Gupta, Geeta Gujral and Neha Aggarwal [11], present a method using a filter to locate the prime areas of the image to hide the information in, but only in the blue component of the image. The authors state that it improves the performance of LSB by only hiding it in one of the three layers, and causes less degradation by only modifying one of the colour values.

Champakamala .B.S, Padmini.K, and Radhika .D. K [12], present a method of embedding information using the LSB-1 and LSB-2. The information is also embedded in reverse order, so instead of '123456', it is embedded as '654321' starting from LSB-2.

Algorithm

The algorithm for my implementation of LSB watermarking is as follows:

Embedding

1. Get the host image
    a. Calculate the number of rows and columns in host image
2. Get the watermark image
    a. Calculate the number of rows and columns in watermark image
3. Let user select the bit plane for the watermark to be embedded into on the host image. (1 = LSB, 8 = MSP (Most Significant Bit))
4. Check watermark size compared to host image size
    a. If equal, proceed to step 5
    b. If watermark is larger than host, resize to same size

c. If watermark is smaller than host, then tile watermark to the same size as host

5. Convert watermark to binary image
6. Combine the host image and the watermark image on the bit plane that user selected
7. Return watermarked image

Recovery

1. Using the known bit plane, extract values for this bit plane value into a new image
2. Return the recovered watermark

Results

As shown in Figure 3 the effects of watermarking an image on the LCB-1 plane are not visible to the human eye. Figure 4 shows the original watermark that was then converted to a binary image (black and white pixels only), and then followed by the recovered watermark. This recovered watermark is identical to the original binary watermark. It is identical, as no transformations have been performed on the image, thus it was able to retrieve the perfect watermark.

Fig. 3. Original image on left, watermarked image on right

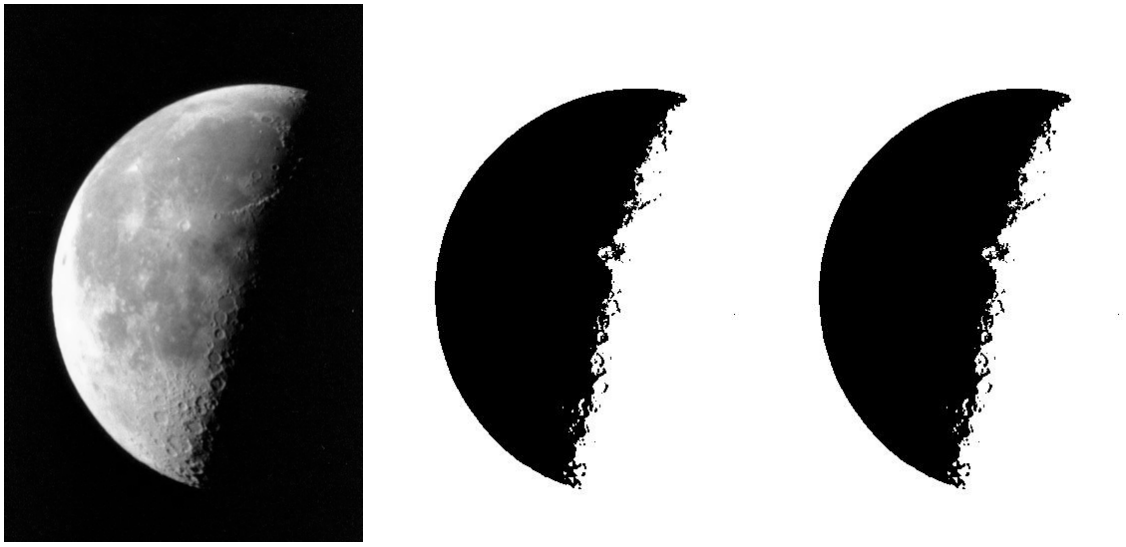

Fig. 4. Original watermark on left, binary version in middle, recovered watermark on left

However, Figure 4 shows what happen when noise is added to the equation. This process involved embedding the watermark, and then adding Gaussian white noise. Next I attempted to recover the watermark using the same algorithm tried prior. This was attempted with four different bit plane levels. The first row is with bit plane one; the recovered watermark is completely unrecognizable. With the second row with a bit plane of four, the watermark is still mostly unrecognizable. The third and fourth rows are at bit planes six and eight, respectfully, and in these cases the watermark is actually recognizable; with bit plane eight being the clearest recovery. This is because Gaussian white noise largely affects the lower bit plane values and leaves the higher ones alone. Also notice how as the bit plane level gets higher the watermark becomes more and more visible in the watermarked image.

The PSNR value was 51.1396 for the images in Figure 3 and Figure 4. While the SNR value was 45.5572. Following that the MSE value was 0.5002. These results are as expected, with a high PSNR and a low MSE.

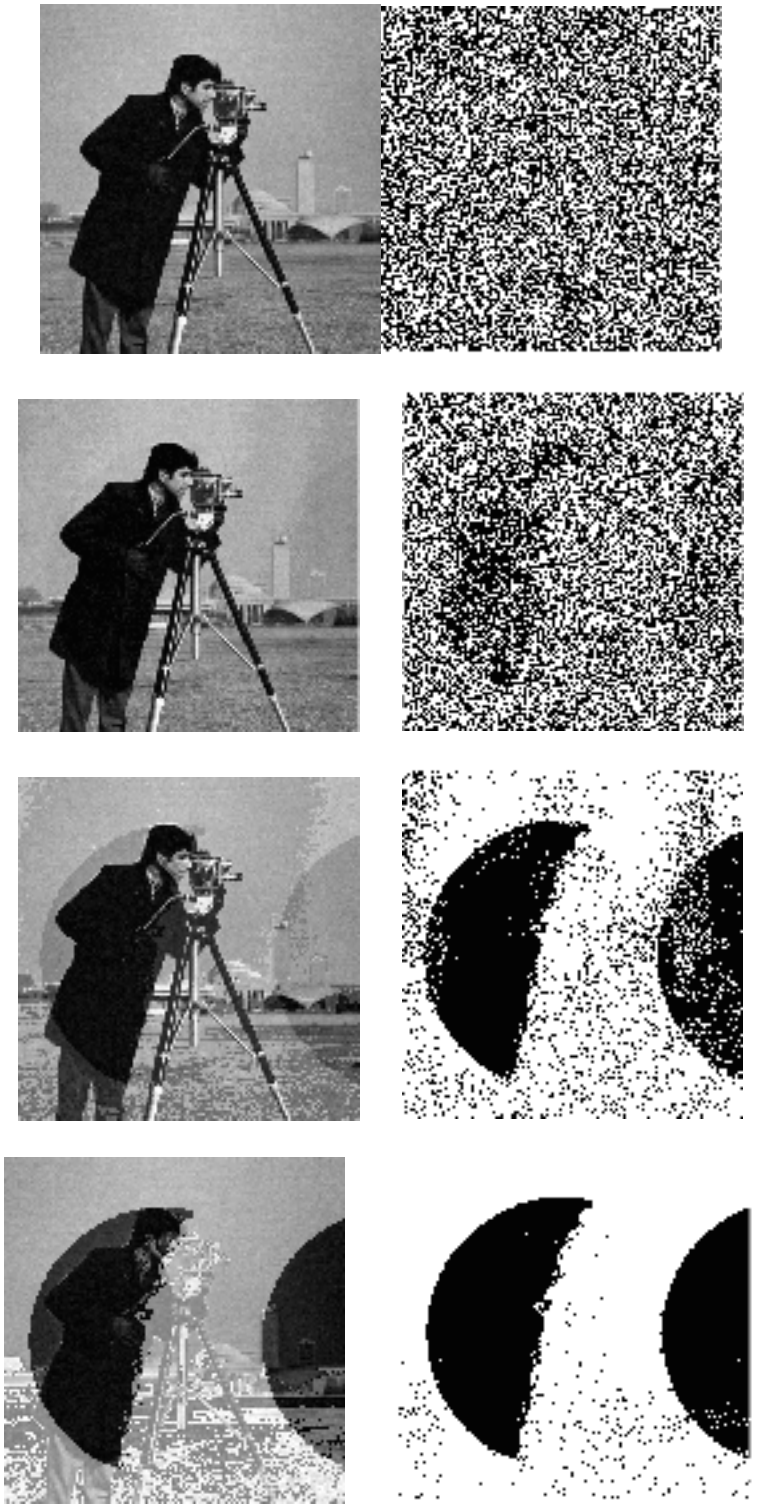

Fig. 5. Left is watermarked image, right is recovered watermark for different levels of bit plane modification, from top to bottom the levels are as follows 1, 4, 6, 8.

Conclusion

Based upon the observations of embedding watermarks on different bit plane levels, it can be clearly stated that the lower the bit plane level, the more fragile a digital watermark is. Whereas the higher the bit plane level the watermark is embedded in, the more robust a digital watermark is, though this does come at a cost as the watermark becomes more and more visible as the bit plane level gets larger. The different methods of LSB implementation discussed in Related Work can clearly come in very handy; as reading a watermark for any image watermarked simply using the standard LSB method is simple. This would also allow anyone to go in, remove the watermark and add a new watermark.

References

[1] Samir El-Seoud and Islam Taj-Eddin, "On the Information Hiding Technique Using Least Significant Bits Steganography," *Int. J. Comput. Sci. Inf. Secur.*, vol. 11, no. 11, pp. 34–45, 2013.

[2] Puneet Kr Sharma and Rajni, "Analysis of Image Watermarking using Least Significant Bit Algorithm," *Int. J. Inf. Sci. Tech.*, vol. 2, no. 4, pp. 95–101, Jul. 2012.

[3] Bank of Canada, "Security," *Bank of Canada*, 2015. [Online]. Available: http://www.bankofcanada.ca/banknotes/bank-note-series/polymer/security/. [Accessed: 24-Apr-2015].

[4] Gurpreet Kaur and Kamaljit Kaur, "Implementing LSB on Image Watermarking Using Text and Image," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 8, pp. 3130–3134, Aug. 2013.

[5] A. E. Mustafa, A. M. F. ElGamal, M. E. ElAlmi, and Ahmed. BD, "A Proposed Algorithm For Steganography In Digital Image Based on Least Significant Bit," *Res. J. Specif. Educ. Fac. Specif. Educ. Mansoura Univ.*, no. 21, pp. 751–767, Apr. 2011.

[6] Herodotus, T. Holland, and P. Cartledge, *The Histories*. New York: Viking, 2014.

[7] S.-H. Liu, H.-X. Yao, W. Gao, and Y.-L. Liu, "An image fragile watermark scheme based on chaotic image pattern and pixel-pairs," *Appl. Math. Comput.*, vol. 185, no. 2, pp. 869–882, Feb. 2007.

[8] The MathWorks, Inc., "PSNR," *MATLAB Documentation*, 2015. [Online]. Available: http://www.mathworks.com/help/vision/ref/psnr.html. [Accessed: 25-Apr-2015].

[9] Abdullah Bamatraf, Rosziati Ibrahim, and Mohd. Najib Mohd. Salleh, "A New Digital Watermarking Algorithm Using Combination of Least Significant Bit (LSB) and Inverse Bit," *J. Comput.*, vol. 3, no. 4, Apr. 2011.

[10] Amit Singh, Susheel Jain, and Anurag Jain, "Digital Watermarking Method Using Replacement of Second Least Significant Bit (LSB) with Inverse of LSB," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 2, pp. 121–124, Feb. 2013.

[11] Shilpa Gupta, Geeta Gujral, and Neha Aggarwal, "Enhanced Least Significant Bit algorithm For Image Steganography," *Int. J. Comput. Eng. Manag.*, vol. 15, no. 4, pp. 40–42, Jul. 2012.

[12] Champakamala .B.S, Padmini.K, and Radhika .D. K, "Least Significant Bit algorithm for image steganography," *Int. J. Adv. Comput. Technol.*, vol. 3, no. 4, pp. 34–38, Aug. 2014.