Steganalysis for Reversible Data Hiding

Ho Thi Huong Thom[1] , Ho Van Canh[2] , Trinh Nhat Tien[3]
[1]*Faculty of Information Technology, Hai Phong Private University, Vietnam*
thomhth@hpu.edu.vn
[2]*Dept. of Professional Technique, Ministry of Public Security, Vietnam*
hovancanh@gmail.com
[3]*College of Technology, Vietnam National University, HaNoi, Vietnam*
tientn@vnu.edu.vn

Abstract

In recent years, several lossless data hiding techniques have been proposed for images. Lossless data embedding can take place in the spatial domain or in the transform domain. They utilized characteristics of the difference image or the transform coefficient histogram and modify these values slightly to embed the data. However, after embedding message bits these steganography changed the nature of the difference image histogram or the transform coefficient histogram gradually. In this paper, we introduce two new steganalytic techniques based on the difference image histogram and the transform coefficient histogram. The algorithm can not only detect existence of secret messages in images which are embedded by above methods reliably, but also estimate the amount of hidden messages exactly.

Keywords: *Steganography, Steganalysis, Cover Image, Stego Image, The Difference Image, Histogram Shifting, Lossless Data Hiding, Integer Wavelets*

1. Introduction

Steganography is the art of secret communication, its purpose is to convey messages secretly by concealing the very existence of the message. Similar to cryptanalysis, steganalysis attempts to defeat the goal of steganography. It is the art of detecting the existence of hidden information. Digital images, videos, sound files and other computer files that contain perceptually irrelevant or redundant information can be used as "cover" or carriers to hide secret messages. After embedding a secret message into the cover-image, a so-called stego-image is obtained.

In recent years, several lossless data hiding techniques have been proposed for images. Lossless data embedding can take place in the spatial domain [1, 2, 3], or in the transform domain [4, 5]. Lee et al [3] proposed a lossless data embedding technique (we assume that the technique name DIH method), which utilizes characteristics of the difference image and modifies pixel values slightly to embed the secret data. Xuan et al [5] proposed a histogram shifting method in integer wavelet transform domain (we assume that the technique name IWH method). This algorithm hides message into integer wavelet coefficients of high frequency subbands.

In this paper, we propose two new steganalytic methods based on the difference image histogram and integer wavelets transform. Former can detect stego images using Lee's steganography (DIH method) and latter detects stego images using Xuan's method (IWH method). The algorithms can also estimate the embedded data length reliably. In the next

section, we describe again the details of Lee's and Xuan's steganography. In section 3, we present our proposed steganalytic methods. Our experimental results are given in section 4. Finally, concluding in section 5.

2. Reversible Data Hiding

2.1 Lossless Data Hiding Based on Histogram Modification of Difference Image

In this subsection, we describe again details of Lee and his colleague's lossless data hiding method using the histogram modification of the difference image [3].

2.1.1. Watermark Embedding

They assume that embedded data is a binary Logo sequence B(m,n) of size PxQ pixels, they combine a binary random sequence generated by the user key A(l) of length PxQ bits with B(m,n) using the bit – wise XOR operation, they get a binary watermark sequence W(m,n) of size PxQ.

For a grayscale image I(i, j) of size MxN pixels, they form the difference image D(i, j) of size MxN/2 from the original image. D(i, j) = I(i, 2j+1)-I(i,2j), $0 \le i \le M-1$, $0 \le j \le N/2-1$ where I(i, 2j+1), I(i,2j) are the odd-line field and the even-line field, respectively.

For watermark embedding, they empty the histogram bins of -2 and 2 by shifting some pixel values in the difference image. If the difference value is greater than or equal to 2, they add one to the odd-line pixel. If the difference value is less than or equal to -2, they subtract one from the odd-line pixel. Then, the difference image is modified $\tilde{D}(i,j) = \tilde{I}(i, 2i+1) - \tilde{I}(i, 2j)$ where $\tilde{I}(i, 2j+1)$ and $\tilde{I}(i,2j)$ are the odd-line field and the even-line field of the modified image, respectively.

In the histogram modification process, the watermark W(m,n) is embedded based on the modified difference image $\tilde{D}(i,j)$. The modified difference image is scanned. Once a pixel with the difference value of -1 or 1 is encountered, they check the watermark to be embedded. If the bit to be embedded is 1, they move the difference value of -1 to -2 by subtracting one from the odd-line pixel or 1 to 2 by adding one to the odd-line pixel. If the bit to be embedded is 0, they skip the pixel of the difference image until a pixel with the difference value -1 or 1 is encountered. In this case, there is no change in the histogram. Therefore, the watermarked fields $I_w(i, 2j+1)$ and $I_w(i,2j)$ are obtained by

$$I_w = \begin{cases} \tilde{I}(i,2j+1)+1 & \text{if } \tilde{D}(i,j)=1 \text{ and } W(m,n)=1 \\ \tilde{I}(i,2j+1)-1 & \text{if } \tilde{D}(i,j)=-1 \text{ and } W(m,n)=1 \\ \tilde{I}(i,2j+1) & \text{otherwise} \end{cases} \quad (1)$$

and

$$I_w(i,2j)=I(i,2j). \quad (2)$$

2.1.2. Watermark Extraction and Recovery

Calculating the difference image $D_e(i,j)$ from the received watermarked image $I_e(i,j)$. The whole difference image is scanned. If the pixel with the difference value of -1 or 1 is

encountered, the bit 0 is retrieved. If the pixel with the difference value of -2 or 2 is encountered, the bit 1 is retrieved. In this way, the embedded watermark $W_e(m,n)$ can be extracted.

Finally, we reverse the watermarked image back to the original image by shifting some pixel values in the difference image. The whole difference image is scanned once again. If the difference value is less than or equal to -2, they add one to the odd-line pixel. If the difference value is greater than or equal to 2, they subtract one from the odd-line pixel.

2.1.3. Lossless Image Recovery

The proposed scheme cannot be completely reversed because the loss of information occurs during addition and subtraction at the boundaries of the grayscale range (at the gray level 0 and 255). In order to prevent this problem, they adopt modulo arithmetic for watermark addition and subtraction. For the odd –line field I(i,2j+1), they define the addition modulo c as

$$I(i,2j+1) +_c 1 = (I(i,2j+1)+1) \bmod c \tag{3}$$

where c is the cycle length. The substraction modulo c is defined as

$$I(i,2j+1) -_c 1 = (I(i,2j+1)-1) \bmod c \tag{4}$$

The reversibility problem arises from pixel that is truncated due to overflow or underflow. Therefore, they use $+_c$ and $-_c$ instead of + and – only when truncation due to the occurrence of overflow or underflow. In other words, they have only to consider $255 +_c 1$ and $0 -_c 1$.

In the receiving side, it is necessary to distinguish between the case when, for example, $I_e(i, 2j+1)=255$ was obtained as $I(i,2j+1)+1$ and $I(i,2j+1) -_{256} 1$. They assume that no abrupt change between two adjacent pixels occurs. If there is a significant difference between $I_e(i, 2j+1)$ and $I_e(i, 2j)$, we estimate that I(i,2j+1) was manipulated by modulo arithmetic.

$$\begin{cases} I(i,2j+1)+1 & \text{if } |I_e(i,2j+1) - I_e(i,2j)| \le \tau \\ I(i,2j+1) -_{256} 1 & \text{otherwise} \end{cases} \tag{5}$$

Where τ is a threshold value. Similarly, $I_e(i,2j+1)=0$ is estimated as

$$\begin{cases} I(i,2j+1)-1 & \text{if } |I_e(i,2j+1) - I_e(i,2j)| \le \tau \\ I(i,2j+1) +_{256} 1 & \text{otherwise} \end{cases} \tag{6}$$

2.2. Lossless Data Hiding Based on Integer Wavelet Histogram Shifting

In the subsection, we describe again the detail of IWH method. Since it is required to reconstruct the original image with no distortion, Xuan et al [5] use the integer lifting scheme wavelet transform. After integer wavelet transform, it has four sub-bands. They will embed the information into three high frequency sub-bands. IWH method is presented as follows.

2.2.1. Data Embedding Algorithm

Assume there are M bits which are supposed to be embedded into a high frequency subband of IWT. We embed the data in the following steps:

(1). Let a threshold T>0 be the number of the high frequency wavelet coefficients in [-T, T] is greater than M. And set the Peak=T.

(2). In the wavelet histogram, move the histogram (the value is greater than Peak) to the right-hand side by one unit to leave a zero-point at the value Peak+1. Then embed data in this point. Scanning all of IWT coefficients in the high frequency subband. Once an IWT coefficient of value "Peak" is encountered, if the to be embedded bit is 1, this coefficient's value will be added by 1, i.e, becoming "Peak+1". If the to be embedded bit is 0, the coefficient's value remain to be "Peak".

(3). If there are to-be-embedded data remaining, let Peak = (-Peak), and move the histogram (less than Peak) to the left-hand side by 1 unit to leave a zero-point at the value (-Peak-1). And embed data in this point.

(4). If all the data are embedded, then stop here and record the Peak value as stop peak value, S. Otherwise, Peak =(-Peak-1), go back to (2) to continue to embed the remaining to-be-embedded data.

2.2.2. Data Extraction Algorithm

Data extraction is the reverse process of data embedding. Assume the stop peak value is S, the threshold is T. The data extraction process is performed as follows:

(1). Set Peak = S.

(2). Decode with the stop value Peak. (In what follows, assume Peak>0). When an IWT coefficient of value "Peak+1" is met, bit "1" is extracted and the coefficient's value reduces to "Peak". When the coefficient of value "Peak" is met, bit "0" is extracted. Extract all the data until Peak+1 becomes a zero-point. Move all the histogram (greater than Peak+1) to the left-hand by one unit to cover the zero-point.

(3). If the extracted data is less than M, set Peak= -Peak. Continue to extract data until it becomes a zero-point in the position (Peak-1). Then move histogram (less than Peak-1) to the right-hand side by one unit to cover the zero-pint.

(4). If all the hidden bits have been extracted, stop. Otherwise, set Peak=-Peak+1, go back to (2) to continue to extract the data.

3. Proposed Steganalytic Methods

3.1. Steganalytic Method for DIH method.

After embedding a set of message into a set of original image using DIH method get a set of stego image, we form the difference image D(i,j) and calculate the histogram of D(i,j) for each image in the original image set and the stego image set, we found out that DIH method changed natural of the difference image histogram of typical image significantly as the example in [3].

We use a original Lena grayscale image of size 512 x 512 pixels (Figure 1. (a)), we perform the difference image D(i,j) and calculate histogram of all D(i,j) that is shown in Figure 1. (c). We then embed a watermark which is a binary logo image of size 128x56 pixels, equivalent to a binary sequence of 7,168 bits (following the example of [3], see Fig.1. (b)) into Lena image using DIH. We get the difference image histogram in Figure 1. (d).

From Figure.1 (c) and (d), comparing difference between the original histogram and the histogram after data embedding we see easily that DIH changed the difference value of -2 and 2 considerably. In any typical image, the histogram value of the difference value -2 and 2 (denote h_2 and h_{-2}) is always greater than the histogram value of the difference value -3 and 3 (denote h_3 and h_{-3}), it mean that sum of h_2 and h_{-2} greater than sum of h_3 and h_{-3} τ time ($\tau \geq 1$ is a threshold). In other words, in a stego image, the sum of h_2 and h_{-2} is less than of h_3 and h_{-3} τ time.

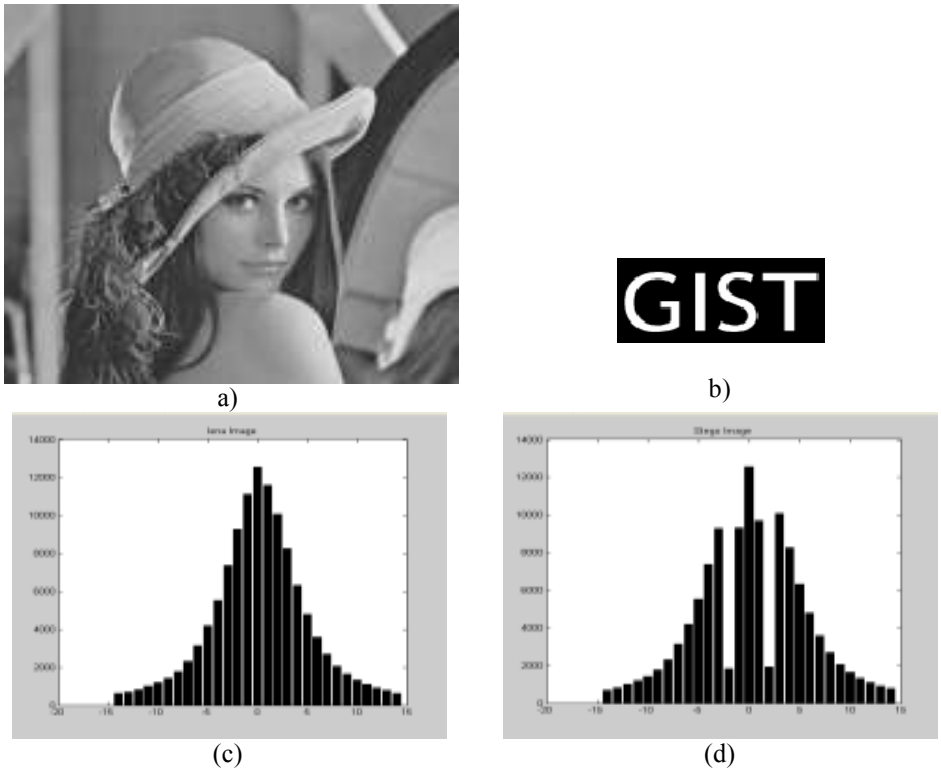


Figure.1. Test images: a) Lena original image, b) Binary logo image. (c) The difference image histogram of original image, (d) The difference image histogram of stego image.

In certain cases, the sum of h_2 and h_{-2} is less than of h_3 and h_{-3} τ time, we can conclude that this image is a stego image using DIH method and we can also estimate data length which was embedded in original image as the following analysis.

We assume that the to be embedded data is a binary sequence W of n bits with the number of bit "1" equals to the number of bit "0", approximately. After embedding the hidden sequence into an original image, DIH method will shift a part n/2 of h_1 and h_{-1} to h_2 and h_{-2} to store n/2 message bits, it means a part n/2 of the original h_1 and h_{-1} now becomes n/2 of h_2 and h_{-2}, remaining n/2 data bits are embedded into a part n/2 of h_1 and h_{-1}. Therefore, the

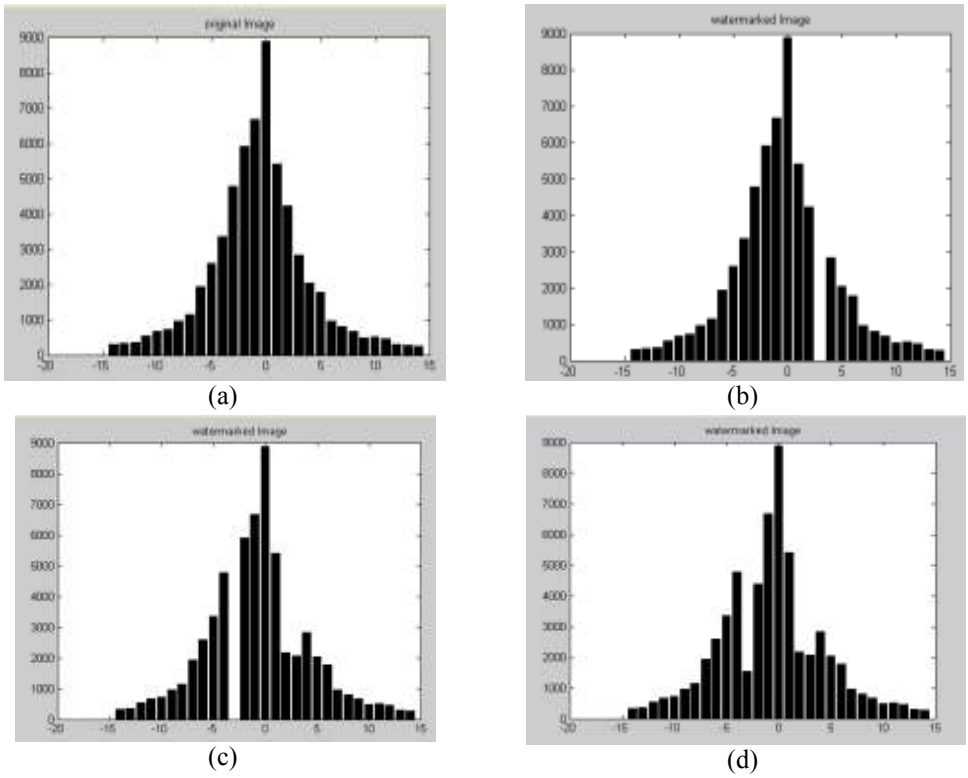
hidden data length equals to $(h_2+h_{-2})*2$, approximately. In our experimentation, we get high reliable result with $\tau = 1.15$.

Applying to above example, we estimate the embedded data length L=7034 bits which are embedded in Lena image.

In special case, users doesn't use different value of -2 and 2 to embed data, they choose other different values. We change the method a little. The difference image histogram is scanned. Once a pair of histogram of i+1 and – (i+1) is greater than of i and –i with τ time, we set i be the location to estimate the embedded data length.

3.2. Steganalytic Method for IWH Method

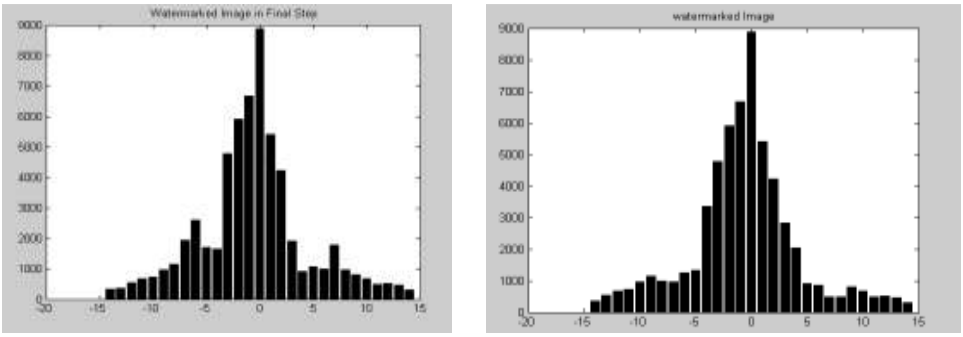
To estimate message length in stego image using IWH method, we first give analysis of occurrences in watermarking process as the three following experiments:



(a) (b)

(c) (d)

Figure 2. An example showing how a zero point is generated and payload data embedding process: (a) original histogram, (b) histogram after a zero point is created, (c) histogram after data embedding at Peak =2 and then a new zero point is created at new next Peak, (d) histogram after data remaining embedding with new Peak.



(a) (b)

Figure 3. Another example showing how payload data embedding process: (a) the histogram after data embedding with chosen T=4, (b) histogram after data embedding with chosen T=6.

In the first experiment, we use also Lena image and Logo image in section 3.1 to test. After integer wavelet transform, we calculate the histograms of high frequency subbands (see Figure 2. (a)). We next embed payload data (that is the binary sequence from Logo image) in to the high frequency subbands with T=2 using IWH method. We get S=-2 and calculate again the high frequency subbands that is shown in Figure 2. (d). The data embedding process performs via some steps: the first and second step embeds data in the point 2 and 3 (see Figure 2. (b), (c)) but there are to be embedded data remaining, the process performs the third and fourth step with T=-2 to embed data (see Figure 2. (c), (d)). In the second experiment, we use also the Lena original image and Logo watermark with T=4, we then get S=3. In this case, the histogram is changed much that is shown in Figure 3. (a). In the third experiment, we use the same input with T=6, we then get S=-5. In this case, the histogram is changed clearly that is shown in Figure 3. (b).

We compare difference between the histogram of typical image and of stego image, we found out that, in typical image, $h_0>h_1> h_2 > h_3>\ldots$ and $h_0>h_{-1}>h_{-2}>h_{-3}>\ldots$ where h_i is histogram value of integer wavelet coefficient i.

The stego image in the first experiment, we get $h_4 >h_3$, $h_3 \approx h_2$, $h_{-4} > h_{-3}$, $h_{-3} < h_{-2}$.

The stego image in the second experiment, we get $h_5 \approx h_6$, $h_{-5} \approx h_{-4}$, $h_4 < h_3$, $h_4 <h_5$

The stego image in the third experiment, we get $h_7 \approx h_8$, $h_5 \approx h_6$, $h_{-7} \approx h_{-8}$, $h_{-5} \approx h_{-6}$.

We explain detail of these problems of the third experiment. IWT method first shift the part of histogram with value greater than T=6 towards the right hand side by one unit. After data embedding we get $h_6 \approx h_7$. Due to data remaining, Peak T=6 change new T=-6, at this step, amount of remaining data fit histogram value -6, so after data embedding, $h_{-6} \approx h_{-7}$. Next, T=-6 becomes T=5, h_6 and h_7 move to h_7 and h_8, and $h_5 \approx h_6$ due to remaining available payload. T =5 change to T=-5 again, h_{-6} and h_{-7} move to h_{-7} and h_{-8} to embed data, remaining payload embed into a part of h_{-5}, it makes a part of h_{-5} become h_{-6} (due to a number of remaining data equal h_{-5}, so $h_{-5} \approx h_{-6}$). Finally, IWT method finish and set S=T=-5.

Table 1. Experimental result on Lena image

Embedded data length	Chosen threshold T	Gotten Stop value S	Estimated data length
7168	2	-2	7231
7168	4	3	6998
7168	6	-5	7177

From above analyses, we give generally steganalytic algorithm estimating length of payload data as follows:

(1). Initiate data length L=0, scan all histogram of value i (i>=0, i <=max (all integer wavelet coefficient of high subbands)), if the first $(h_i + h_{i+1})/2 < h_{i+2}$ is met, stop scanning, let Peak = i be first location to estimate data length.

(2). if $h_{Peak} \approx h_{Peak+1}$, $L=L+h_{Peak}+h_{peak+1}$; set Peak = -Peak and perform next step 3. Otherwise, perform step 4.

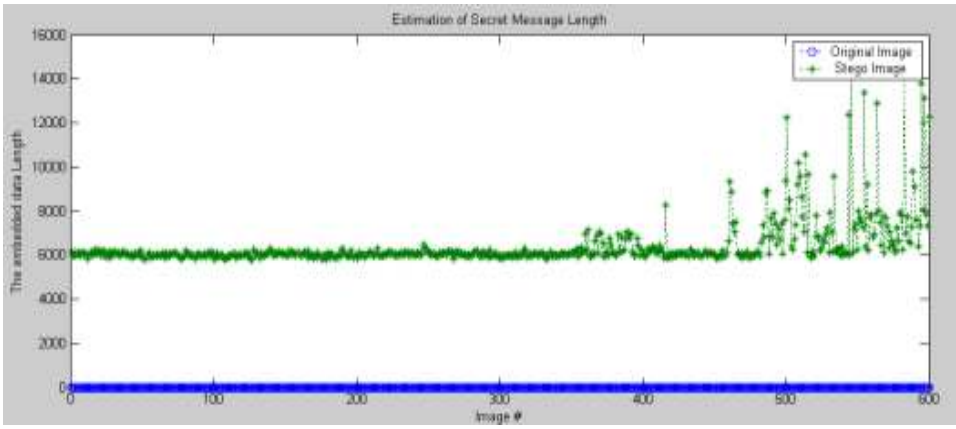
(3). if $h_{Peak} \approx h_{Peak+1}$, $L=L+h_{Peak}+h_{peak+1}$; set Peak = -Peak - 1 and return step 2. Otherwise, perform step 4.

(4). if $h_{Peak+1}<h_{Peak+2}$ and $h_{Peak+1}< h_{Peak}$ then $L= L+2*h_{Peak+1}$. The process finishes here.

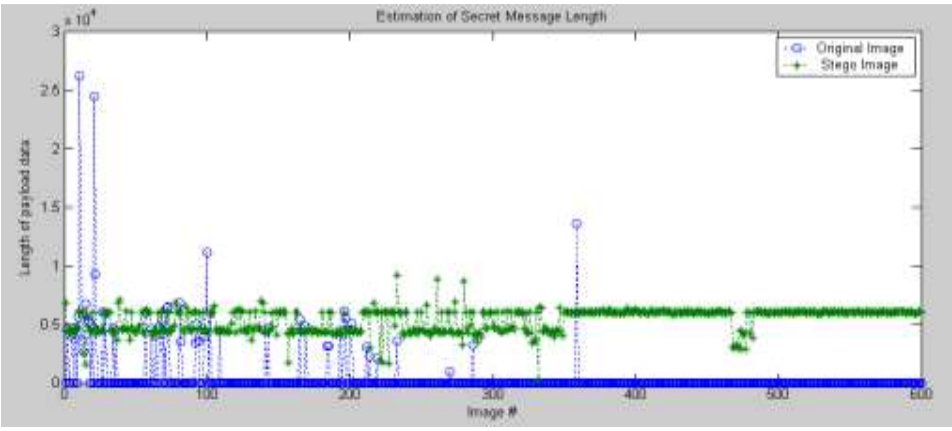
Applying the algorithm to three above experiments, we estimate the embedded data lengths that are shown in table 1.

4. Experimental Results

We have a set of images, it includes 600 grayscale images with 350 standard test JPEG image size of 768x512 or 512x512 pixels they were downloaded from [6], [7], and 250 test JPEG images with 1280x960 pixels they were created from my digital camera, all images are then converted to grayscale images by Photoshop CS2 software.



(a)



(b)

Figure 4. Experimental results: (a) Estimated message length for the database using DIH method, (b) Estimated message length for the database using IWH method

From the above set, we create two new sets. First set includes 1200 images with 600 original images and 600 stego-images which are embedded the same secret binary sequence of 6000 bits into corresponding 600 original images by DIH method. Second set includes also 1200 images with 600 original images and 600 stego-images which are embedded the same secret binary sequence of 6000 bits into corresponding 600 original images by IWH method. Then, we use our two proposed steganalytic methods to detect cover image and stego image and estimate embedded data length for two sets, respectively. The test results are shown in Figure. 4. (a) and (b). There the horizontal axis represents image number # and the vertical axis represents the embedded data length corresponding image number #.

In the first experimental result, our first steganalytic method detected exactly 600 original images (100 %) and the mean of estimated data lengths equals to 6397.5. In the second experimental result, our second steganalytic method detects exactly 544 original images (90.67 %) and the mean of estimated data lengths equals to 5356.5.

Comparing accuracy between two methods we found that error estimation concentrates on very noisy images. So we give several factors that influence the accuracy of the estimation: **Noise:** For very noisy images, it makes histogram value of difference image and high integer wavelet coefficient become closer. So our steganalytic methods can detect falsely. **Chosen Peak:** If T is chosen be greater than 10 for IWH method, our method 2 will be very hard to estimate embedded data length in stego image.

5. Conclusions

This paper proposes two new steganalytic algorithms, which bases on histogram of the difference image and integer wavelet coefficient of high subbands. Experimental results show that our methods are reliable. However, it is hard to detect stego-image with two factors which are shown in section 4. We acknowledge that there are many elements in our algorithms that can be changed or replaced with other elements.

References

[1]. Honsinger, C., Jone, P., Rabbani, M., Stoffel, J.: Lossless recovery of an original image containing embedded data. US Patent: 6,278,791 B1 (2001).

[2]. Ni, Z., Shi, Y., Ansari, N., Su, W.: Reversible data hiding. Proc. ISCAS (2003), pp 912–915.

[3]. Sang-Kwang Lee, Young-Ho Suh, and Yo-Sung Ho, Lossless Data Hiding Based on Histogram Modification of Difference Images, PCM 2004, LNCS 3333 (2004), pp 340–347.

[4]. Xuan, G., Zhu, J., Chen, J., Shi, Y., Ni, Z., Su, W.: Distortionless data hiding based on integer wavelet transform. IEEE Electrionics Letters (2002), pp 1646–1648.

[5]. Guorong Xuan, Qiuming Yao, Chengyun Yang, Jianjiong Gao, Peiqi Chai, Yun Q. Shi, Zhicheng Ni, Lossless Data Hidding Using Histogram Shifting Method Based on Integer Wavelets, Proc. 5th Digital watermarking workshop, IWDW 2006, Korea, vol. 4283, pp. 323-332.

[6]. CBIR image database, University of Washington, available at: http://www.cs.washington.edu/research/imagedatabase/groundtruth/

[7]. USC-SIPI Image Database, http://sipi.usc.edu/services/database/Database.html.

Authors

Ho Thi Huong Thom received the B.S. degree of Information Technology department from Haiphong Private University and the M.S. degree in Information Systems from College of Technology, Vietnam National University in Vietnam, in 2001 and 2005, respectively. She has started her career as Lecturer in Department of Information Technology in Haiphong Private University, Vietnam and served for 9 years. Currently, she is pursuing Doctor of Information Systems from College of Technology, Vietnam National University, Hanoi, Vietnam. Her research interests includes Image processing, Information Security, Information Hiding.

Ho Van Canh received the B.S. degree in Mathematics from Hanoi City University in Vietnam in 1973, the Dr. Sci. degree in Faculty of statistology from KOMENSKY University in Czechoslovakia in 1987. Currently, he has been working as a cryptologist in Dept. of Professional Technique, Ministry of Public Security, Vietnam. His research interests include cryptography, information security, information hiding.

Trinh Nhat Tien received the B.S degree from University of Prague in Czechoslovakia in 1974, and the Dr. degree from University of Prague, Czechoslovakia and University of Hanoi, Vietnam in 1984. He has started as Lecturer in Department of Information Technology of College of Technology, Vietnam National University, Hanoi, Vietnam since 1974. His research interests include algorithm, complexity of algorithm, information security.