低失真的最低有效位元匹配法

曾顯文
朝陽科技大學資管系
e-mail:
hwtseng@cyut.edu.tw

李慧芳
朝陽科技大學資管系
e-mail:
s9614611@cyut.edu.tw

摘要

在本篇論文中，我們提出一個低失真的最低有效位元匹配(Least-Significant-Bit Matching, LSB Matching) 的隱藏學方法。2006 年 Mielikainen 學者改進了傳統的 LSB Matching，這個方法是以二個像素為單位，一次藏入二個機密訊息，每次藏入只需修改一個像素值，因此，使得藏入時的像素修改量減少，以提高偵測的難度與影像的品質。本方法則是提出以三個像素為單位，每次藏入三個機密訊息，藏入的方法與 LSB Matching 相似，因此，本方法的藏入量與 Mielikainen 的方法相同，但是藏入時的像素修改率降低，並提高影像品質。在實驗結果中，本方法將與 Mielikainen 的方法進行比較，證明所提出的方法效能較佳，失真度較低。

關鍵詞：資訊隱藏、隱藏學、最低有效位元匹配。

Abstract

In this paper, we proposed a low distortion least-significant-bit (LSB) matching, steganographic method. In 2006, a modified LSB Matching was proposed by Mielikainen. The method used a pair of pixels as an embedding unit. Two secret bits are embedded into the unit by incrementing or decrementing a corresponding pixel value. Thus the altering rate of pixel value is reduced and the image quality is enhanced. Our proposed method uses three pixels as an embedding unit, three secret bits are embedded into the unit. The embedding method is similar with Mielikainen's LSB matching method. Therefore, the embedding payload of the proposed method is the same as that of Mielikainen's method. However, the altering rate of pixel value can be more reduced and the image quality of stego image can be more enhanced. The experimental results show that the proposed method is superior to Mielikainen's method in terms of distortion.

Keywords: Data hiding, steganography, LSB matching.

1. 前言

數位資訊在網際網路中傳遞、交換時容易受到第三者的攻擊、攔截，為了保護數位資訊，資訊隱藏是必需的。而隱藏學(Steganography)[1,3-7,13-14]是一種資訊隱藏技術的應用，將機密訊息藏到遮蓋媒體(Cover Media)中，以產生隱蔽媒體(Stego-Media)，因為隱蔽媒體和遮蓋媒體非常相似，所以當隱蔽媒體在傳輸的過程中，不會引起其他人的懷疑。一個好的資訊隱藏技術必須滿足安全性(Security)、不可察覺性(Imperceptibility)、高資訊負載量(Payload)的特性。

一般最為常見的資訊隱藏技術為最低有效位元取代法(LSB replacement method)[3]，最低有效位元取代法是將機密訊息位元逐一的藏入遮蓋影像(Cover image)像素中，若機密訊息跟遮蓋影像像素的 LSB 不同時，則以機密訊息直接取代像素的 LSB，得到一張隱蔽影像(Stego image)，達成資訊隱藏的目的。然而最低有效位元取代法的藏入動作讓原本的影像產生不平衡的現象，很容易被偵測[2,8-12,15]及取出藏入的資訊。因此，提出了最低有效位元匹配(LSB Matching)[1]改進了最低有效位取代法的缺點，LSB Matching 的藏入法與最低有效位元取代法不同，當機密訊息跟遮蓋影像像素的 LSB 不同時，是隨機的以增加或減少像素的值，維持整體的像素的平衡。

在 2006 年 Mielikainen 學者提出一個改良 LSB Matching[6]的方法，此方法與傳統的 LSB Matching 的藏入法不同，不是以隨機選取要修改的像素值，而是透過一個二元函式來選擇要修改的像素值，每次藏入二個機密訊息只需修改一個像素值，使得像素修改率降低，提升不可察覺性。

為了再提高隱蔽影像的品質，我們提出了一個低失真的 LSB Matching，使用與 Mielikainen 學者所提出的 LSB Matching 類似的藏入法，每次藏入三個機密訊息通常只需修

改一個像素值，只有在最差情況下需要修改二個像素值，因此更可以維持影像像素值的平衡，使隱蔽影像更難被察覺到藏有機密訊息。

2.相關文獻探討

Mielikainen 學者在 2006 年提出一個改良的最低有效位元匹配隱藏法(Least-Significant-Bit Matching, LSB Matching)[6]，此隱藏法使用於灰階影像上。在資訊藏匿時，以二個像素值為一對，每次藏入二個機密位元，判斷選擇其中一個像素值進行增加或減少像素的值，達到資訊隱藏的目的。由於每次藏匿的過程都只需修改一個遮蓋影像的像素值，因此使用此方法可以減少像素的修改率，使整體的像素維持在一定的平衡上，維持影像品質。

Mielikainen 學者首先將灰階影像中相鄰的兩個像素視為一對，每次藏入相對應的兩個機密訊息位元，藏入的過程中，先判斷第一個像素值的 LSB 是否等於藏入的機密訊息位元，當兩者不同時，使用一個二元函式(Binary Function)公式(1)來選擇要修改的像素值，假設隱蔽影像的二個像素值為 x_i 及 x_{i+1}，藏入的訊息位元為 m_i 及 m_{i+1}，藏入後的影像像素值為 y_i 及 y_{i+1}，其藏入的過程以二元樹表示，如圖 1 所示。

$$f(x_i, x_{i+1}) = LSB(\left\lfloor \frac{x_i}{2} \right\rfloor + x_{i+1}) \qquad (1)$$

例如：假設 $x_1 = 34$ 和 $x_2 = 45$，要藏入的訊息為 $m_1 = 1$ 和 $m_2 = 0$；則判斷 $[LSB(34) = 0] \neq [m_1 = 1]$，$[f(34-1, 45) = 1] \neq [m_2 = 0]$；所以 $y_1 = 35$，$y_2 = 45$。

假設 $x_1 = 34$ 和 $x_2 = 45$，要藏入的訊息為 $m_1 = 0$ 和 $m_2 = 1$；則判斷 $[LSB(34) = 0] = [m_1 = 0]$，$[f(34, 45) = 0] \neq [m_2 = 1]$；$y_1 = 34$，$y_2 = 44$。

在取出機密訊息時，與藏入時的方式相同，將隱蔽影像相鄰的二個像素值為一組 (y_i, y_{i+1})，先將 y_i 的 LSB 取出即可得到機密訊息 m_i，接著利用二元函式 $f(y_i, y_{i+1})$ 公式(2)取出 m_{i+1}。例如：$y_1 = 35$，$y_2 = 45$，得到機密訊息 $m_1 = 1$ 和 $m_2 = 0$。

$$f(y_i, y_{i+1}) = LSB(\left\lfloor \frac{y_i}{2} \right\rfloor + y_{i+1}) \qquad (2)$$

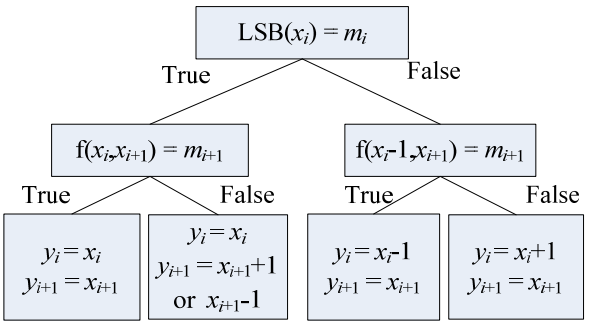

圖 1 LSB Matching 藏入過程示意圖

3.提出的方法

我們所提出的方法是藉由降低像素修改率的概念，修改 Mielikainen 學者所提出的方法。本方法是使用在灰階影像上，將影像依照奇數列及偶數列分別地從左至右及從右至左，由上至下的將影像像素分配以三個像素為一對，如圖 2 所示。每次藏入三個機密訊息，假設隱藏影像的三個像素值 x_i、x_{i+1} 及 x_{i+2}，藏入的機密訊息位元為 m_i、m_{i+1} 及 m_{i+2}，藏入後的影像像素值為 y_i、y_{i+1} 及 y_{i+2}，其藏入的過程與 Mielikainen 學者的方法類似，如圖 3 所示。

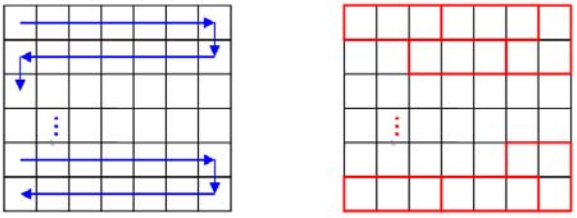

圖 2 影像像素分對圖

3.1 嵌入機密資訊流程

步驟1. 將遮蓋影像以相鄰的三個像素分配成一組，假設為 x_i、x_{i+1} 及 x_{i+2}，將機密訊息也以三個像素分配成一組，假設為 m_i、m_{i+1} 及 m_{i+2}，第一對藏入像素為 (x_{i+2}, x_i)，第二對藏入像素為 (x_i, x_{i+1})，第三對藏入像素則為

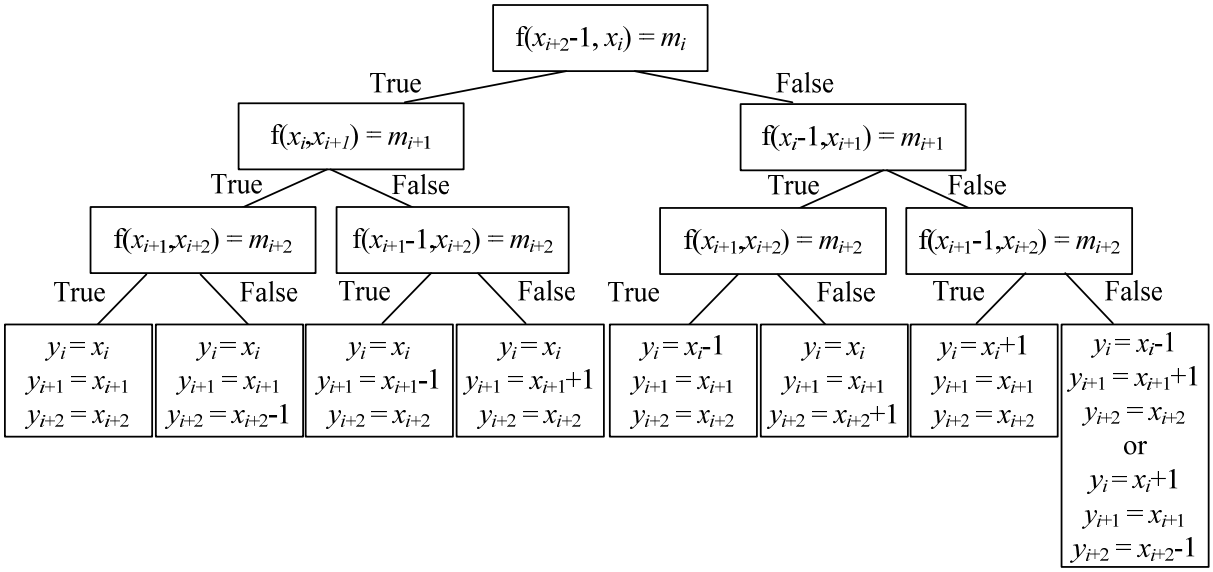

圖 3 藏入過程示意圖

(x_{i+1}, x_{i+2})。首先利用公式(1) 計算 $f(x_{i+2}-1, x_i)$，判斷第一對像素的 LSB 是否等於 m_i。

步驟1. 若 $f(x_{i+2}-1, x_i)$ 等於 m_i，則判斷第二對像素 $f(x_i, x_{i+1})$ 的 LSB 是否等於 m_{i+1}。若不相同，則判斷 $f(x_i-1, x_{i+1})$ 的 LSB 是否等於 m_{i+1}。

步驟2. 若 $f(x_i, x_{i+1})$ 或 $f(x_i-1, x_{i+1})$ 等於 m_{i+1}，則進行判斷第三對像素 $f(x_{i+1}, x_{i+2})$ 的 LSB 是否等於 m_{i+2}。若不相同，則判斷 $f(x_{i+1}-1, x_{i+2})$ 的 LSB 是否等於 m_{i+2}。

步驟3. 依照判斷後的結果，分別進行像素值的修改，如圖 3 所示。

假設 $x_1=35$、$x_2=45$、$x_3=44$ 和 $m_1=1$、$m_2=0$、$m_3=0$。

步驟1. 判斷 $f(x_3-1, x_1)$ 求得：

$f(44-1, 35)=0\neq m_1=1$。(False)

步驟2. 因為 $f(x_3-1, x_1)\neq m_1$ → 判斷 $f(x_1-1, x_2)$ 求得：

$[f(35-1,45)=0]=[m_2=0]$。(True)

步驟3. 因為 $f(x_1-1, x_2)=m_2$ → 判斷 $f(x_2, x_3)$ 求得：

$[f(45,44)=0]=[m_3=0]$。(True)

步驟4. $y_1=(x_1-1)=(35-1)=34$，

$y_2=x_2=45$，$y_3=x_3=44$。

因為藏入的方法以加 1 或減 1 進行像素修改，因此可能會有溢位(Overflow 或 Underflow)的問題產生。當遇到溢位的狀況，可以使用迴圈的方式來進行修改，如圖 4 所示，我們將溢位的情況分成兩種：

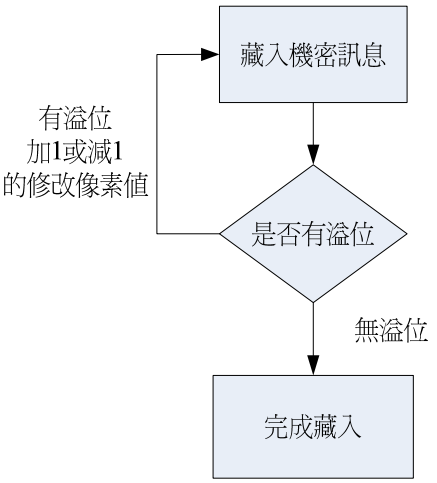

圖 4 循環修改流程圖

機密訊息藏入後，得到 y_i、y_{i+1} 及 y_{i+2} 三個藏入後的影像像素值，接著進行檢測 y_i、y_{i+1} 及 y_{i+2} 是否發生溢位的情況。

狀況1. 上溢(Overflow)：

當溢位的狀況為上溢時，會先將 x_{i+2} 減 1 後，再重新進行機密訊息位元的藏入

程序；藏入後，若仍發生上溢的情況，再繼續將 x_i 減 1，再重新進行藏入的程序；藏入後，若仍發生上溢的情況，再繼續將 x_{i+1} 減 1，再重新進行藏入的程序，直到不會發生溢位的狀況。

例如：假設 $x_1=255$、$x_2=255$ 和 $x_3=255$，要藏入的訊息為 $m_1=1$、$m_2=1$ 和 $m_3=1$；因為 $[f(255-1,255)=0] \neq [m_1=1]$，$[f(255-1,255)=0] \neq [m_3=1]$ 以及 $[f(255-1,255)=0] \neq [m_2=1]$；所以 $y_1=254$、$y_2=256$ 及 $y_3=255$，由此可知 y_2 產生溢位的狀況。

解決方式：先將 x_3 減 1 得到 254 之後，再重新進行機密訊息位元的藏入程序，$x_1=255$、$x_2=255$ 和 $x_3=254$；因為 $[f(254-1,255)=1]=[m_1=1]$，$[f(255,255)=0] \neq [m_2=1]$ 及 $[f(255-1,254)=1]=[m_3=1]$；計算後得到 $y_1=255$、$y_2=254$ 及 $y_3=254$。

狀況2. 下溢(Underflow)：

當溢位的狀況為下溢時，則會先將 x_{i+2} 加 1 後，再重新進行機密訊息位元的藏入程序；藏入後，若仍產生下溢的情況，再繼續將 x_i 加 1，再重新進行藏入的程序；藏入後，若仍產生下溢的情況，再繼續將 x_{i+1} 加 1，再重新進行藏入的程序，直到不會發生溢位的狀況。

例如：假設 $x_1=0$、$x_2=0$ 和 $x_3=0$，要藏入的訊息為 $m_1=0$、$m_2=0$ 和 $m_3=1$；因為 $[f(0-1,0)=0]=[m_1=0]$，$[f(0,\ 0)=0]=[m_2=0]$ 以及 $[f(0,\ 0)=0] \neq [m_3=1]$；所以 $y_1=0$、$y_2=0$ 及 $y_3=-1$，由此可知 y_3 產生溢位的狀況。

解決方式：先將 x_3 加 1 再重新進行機密訊息位元的藏入程序，因此，$x_1=0$、$x_2=0$ 和 $x_3=1$；因為 $[f(1-1,0)=0]=[m_1=0]$，$[f(0,\ 0)=0]=[m_2=0]$ 及 $[f(0,\ 1)=1]=[m_3=1]$；所以 $y_1=0$、$y_2=0$ 及 $y_3=1$。

3.2 取出機密資訊流程

步驟1. 透過公式(2)取出隱蔽影像之像素值的 LSB，即可取得機密訊息。計算 $f(y_{i+2}-1, y_i)$，得到 m_i。

步驟2. 計算 $f(y_i, y_{i+1})$，得到 m_{i+1}。

步驟3. 計算 $f(y_{i+1}, y_{i+2})$，得到 m_{i+2}。

假設 $y_1=34$、$y_2=45$ 及 $y_3=44$。

步驟1. $f(44-1,34)=1$，$m_1=1$。

步驟2. $f(34,45)=0$，$m_2=0$。

步驟3. $f(45,44)=0$，$m_3=0$。

4. 實驗結果

我們實驗使用遮蓋影像包含 Lena、Baboon、Boat、Pepper 四張灰階圖，其影像大小皆為 256x256 及 512x512，如圖 5 所示，機密訊息串流為 Matlab 執行 rand()函式產生，機密訊息串流大小與遮蓋影像尺寸相同。我們藉由計算均方根誤差(mean-square error, MSE)及峰值雜訊比(Peak signal-to-noise ratio, PSNR)來評估隱蔽影像之品質，其公式定義如下列：

1. $MSE=(\frac{1}{w\times h})\sum_{i=1}^{w}\sum_{j=1}^{h}(IM_{ij}-\overline{IM}_{ij})^2$，其中 $w\times h$ 為影像尺寸大小，IM_{ij} 及 $\overline{IM}_{ij}$ 為遮蓋影像像素值及隱蔽影像像素值。

2. $PSNR=10\times\log_{10}(\frac{255^2}{MSE})$ dB。

透過藏入機密訊息位元的程序，我們可推算出每藏入一個位元像素修改的機率，在此假設為 P。當三個像素值的 LSB 與藏入的訊息位元相同時，例如：(True、True、True)，則像素值不需要修改。當有一個像素值的 LSB 與藏入的訊息位元不同時，例如：(True、True、False)、(True、False、True)、(False、True、True)，此時，每三個像素中只需修改一個像素值，因此，有 $\frac{(\frac{3}{8}\times 1)}{3}$ 的機率；當有兩個像素值

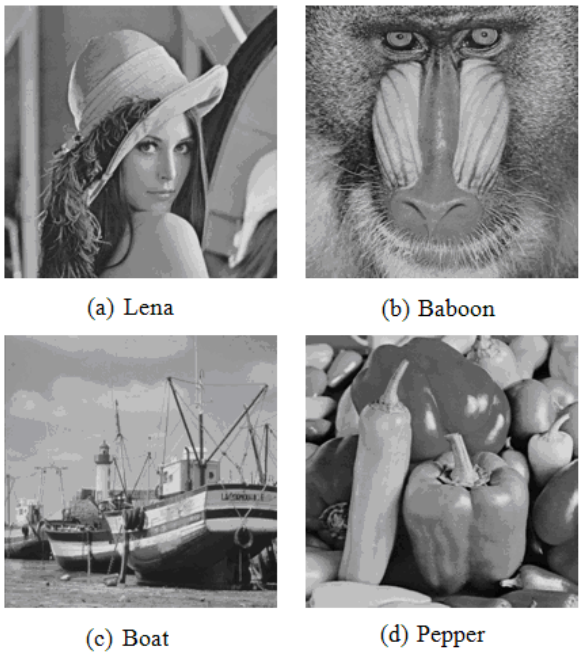

圖 5 實驗灰階影像圖

的 LSB 與藏入的訊息位元不同時，例如：（True、False、False）、（False、False、True）、（False、True、False），此時，每三個像素中也只需修改一個像素值，因此，也有 $\frac{(\frac{3}{8}\times 1)}{3}$ 的機率；最後當三個像素值的 LSB 與藏入的訊息位元不同時，例如：（False、False、False），此時，每三個像素中需要修改兩個像素值，因此，有 $\frac{(\frac{1}{8}\times 2)}{3}$ 的機率。由此可知，我們所提出的方法像素的修改機率 (P) 為 $\frac{(\frac{3}{8}\times 1)+(\frac{3}{8}\times 1)+(\frac{1}{8}\times 2)}{3}=\frac{1}{3}$。

從表 1、表 2 的實驗數據中顯示，雖然我們所提出的方法與 Mielikainen 的方法藏入的資訊量相同，但本篇所提出的方法 PSNR 值都維持在 52.89dB 以上，且像素的修改機率(P)與我們推算出的機率相符，都維持在 0.33 左右，比 Mielikainen 的方法像素的修改機率 0.375 還要低，由此可看出本篇所提出的方法其影像品質較為提昇且像素的修改機率下降。

表 1 256x256 的影像實驗結果

方法 / 影像	Mielikainen 的方法				提出的方法			
	PSNR	MSE	Capacity	P	PSNR	MSE	Capacity	P
Lena	52.3852	0.3755	65536	0.3755	52.9156	0.3323	65536	0.3323
Baboon	52.3810	0.3758	65536	0.3758	52.8987	0.3336	65536	0.3336
Boat	52.3912	0.3749	65536	0.3749	52.8977	0.3337	65536	0.3337
Pepper	52.4013	0.3741	65536	0.3741	52.9071	0.3329	65536	0.3330

表 2 512x512 的影像實驗結果

方法 / 影像	Mielikainen 的方法				提出的方法			
	PSNR	MSE	Capacity	P	PSNR	MSE	Capacity	P
Lena	52.3843	0.3755	262144	0.3755	52.9001	0.3335	262144	0.3335
Baboon	52.3981	0.3743	262144	0.3743	52.8963	0.3338	262144	0.3338
Boat	52.3908	0.3750	262144	0.3750	52.8992	0.3335	262144	0.3336
Pepper	52.3792	0.3760	262144	0.3760	52.8991	0.3336	262144	0.3336

5. 結論

本篇論文所提出的方法可以容易地將機密資訊藏入隱蔽影像中，而且平均每三個像素只修改一個像素值，在傳輸過程中，不容易被第三方發現在隱蔽影像上藏有重要資訊，提高了隱蔽影像被偵測的困難度。在資訊的藏入量方面，雖然我們所提出的方法與 Mielikainen 的方法相同，但在藏入資訊的同一時間內所修改的像素次數較少，降低失真的程度。因此，我們所提出的方法擁有低計算複雜度、高的影像品質及提高不可察覺性。

6. 參考文獻

[1] A. D. Ker, “Improved detection of LSB steganography in grayscale images,” In Proc. Information Hiding Workshop, Vol. 3200, Springer LNCS, pp. 97-115, 2004.

[2] A. D. Ker, “Steganalysis of LSB matching in grayscale images,” IEEE Signal Process. Letters, Vol. 12, No. 6, pp. 441–444, 2005.

[3] C.-K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” Pattern Recognition, Vol. 37, Issue 3, pp. 469-474, 2004.

[4] C.-H. Yang, “Inverted pattern approach to improve image quality of information hiding by LSB substitution,” Pattern Recognition, Vol. 41, Issue 8, pp. 2674-2683, 2008.

[5] A. P. Fabien, R. J. Anderson, and M. G. Kuhn, “Information Hiding – A Survey,” Proceedings of the IEEE Special Issue on Protection of Multimedia Content, Vol. 87, No. 7, pp. 1062-1078, 1999.

[6] J. Mielikainen, “LSB matching revisited,” IEEE Signal Processing Letters, Vol. 13, No. 5, pp. 285-287, 2006.

[7] H.-C. Wu, N.-I. Wu, C.-S. Tsai, and M. -S. Hwang, “Image steganographic scheme based on pixel-value differencing and LSB replacement methods,” Image and Signal Processing, Vol. 152, Issue 5, pp. 611-615, 2005.

[8] R. Chandramouli and N. Memon, “Analysis of LSB based image steganography techniques,” In Proc. IEEE International Conference on Image Processing, Vol. 3, pp. 1019-1022, 2001.

[9] J. Fridrich, M. Goljan and R. Du, “Reliable detection of LSB steganography in color and grayscale images.” In Proc. ACM Workshop on Multimedia and Security. pp. 27-30, 2001.

[10] J. Harmsen and W. Pearlman, “Steganalysis of additive-noise model able information hiding,” In Proc. SPIE Security Watermarking Multimedia Contents, Vol. 5020, pp. 131–142, 2003.

[11] J. Fridrich, M. Golian, and R. Du, “Detecting LSB steganography in color, and gray-scale images,” Multimedia and Security, Vol. 8, Issue 4, pp. 22-28, 2001.

[12] L. Zhi and S.-A. Fen, “Detection of random LSB image steganography,” Vehicular Technology Conference, Vol. 3, pp.2113-2117, 2004.

[13] T. Sharp, “An implementation of key-based digital signal steganography,” In Proc. Information Hiding Workshop, Vol. 2137, Springer LNCS, pp. 13–26, 2001.

[14] X. Zhang and S. Wang, “Steganography using multiple-base notational system and human vision sensitivity,” IEEE Signal Process. Letters, Vol. 12, No. 1 pp. 67-70, 2005.

[15] X. Yu and N. Babaguchi, “An improved steganalysis method of LSB Matching,” In Proc. IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 557-560, 2008.