低失真的資料隱藏方法

A Low Distortion Data Hiding Method

張瑞修
Jui-Hsiu Chang
逢甲大學
資訊工程系
Email:m9689065@fcu.edu.tw

林秀峰
Hsiu-Feng Lin
逢甲大學
資訊工程系
Email:hflin@fcu.edu.tw

陳志瀅
Chih-Ying Chen
逢甲大學
通訊工程系
Email:Chihchen@fcu.edu.tw

Email:m9689065@fcu.edu.tw

***摘要*—在現今資訊科技發達的社會中，使用者享受各種快速、便利的網路服務。但在便利的服務背後，資訊傳輸的安全性卻十分令人憂慮。網路上所有的資料皆以封包的形式進行傳輸與交換。因為封包可能被攔截並進行解讀，所以在網路上傳遞訊息幾乎沒有隱密性可言。為了解決這樣的問題，學者專家們開始研究資料隱藏。資料隱藏可以將訊息偽裝在媒介中，避免資料在傳輸時被人輕易解讀，維護資料傳輸者的隱私。**

本文利用 (5,3) 線性碼的一些性質提出兩種資訊隱藏的方法。方法一是每一個像素可嵌入 3/4 個秘密訊息位元，平均失真率為 0.2344 ，平均 PSNR 值為 54.4726；方法二是每一個像素可嵌入 1 位元的秘密訊息，其平均失真率為 0.3438 及平均 PSNR 值為 52.7677，比用傳統的 LSB 取代法時的平均失真率 0.5 要好。

***關鍵詞*—(5,3) 線性碼、失真率、LSB 取代法**

***Abstract* — Data transferred on the Internet consists of many packets. Anyone with a working knowledge of network operation mechanisms and has the ability to capture packet data can decode it and intercept data. Suffice to say, there is a lack of data privacy on the Internet, and this issue has driven researchers to propose information hiding methods. The information hiding method, Steganography, embeds secret data into a cover image to evade detection and maintain privacy between the sender and receiver. This paper makes use of the property of (5,3) linear code to propose a method of hiding data. The scheme embeds 1 bit of a secret per pixel and the average distortion is 0.3438, with PSNR is 52.7677. The distortion of the proposed methods is better than the LSB substitution method with a distortion of 0.5.**

***Keywords* — (5,3) linear code, distortion, LSB substitution method.**

一、緒論

1. 研究背景

在這個網路發達的時代，可以利用網路做許多事，例如：瀏覽網頁、聊天室、遠距教學、視訊會議、轉播、轉帳、繳稅...等。在做這些事的時候，有許多的資料經由網路來相互傳遞。網路傳遞資料的方式如同郵局一般，電腦將資料切成封包，接著將封包附上接收方與傳送方的 IP 及其他必要的資訊，把封包像包裹一樣交給其他相連的電腦送出去，其他的電腦會檢查收到的封包是不是給自己的 IP ，如果是的話，就把封包留下來，若不是的話，這個封包就會被送給其他人直到這個封包被送到正確的接收者手上。

在網路上使用封包傳遞訊息的時候，若有人在傳送的時候將封包側錄下來拼湊出原來的資料，資料就會外洩，所以網路本身並沒有很好的隱密性。有些封包會利用密碼學的技術達到安全性方面的訴求，即使被拿到也無法破解。但是傳統的加密方法是將資料轉成無意義的亂碼，這反

而會引起怪客(cracker)注意進而嘗試加以破解。

資料隱藏可以將資料偽裝在某種媒介上，避免被第三者監控收送方之間的對話內容，很適合用來解決這種隱密性的問題，因此一直被用在保護機密訊息上。例如傳送方使用資料隱藏將一段文字藏在影像中 (如清明上河圖，奚山行旅圖)，然後將這張偽裝後的影像傳送給收方。惡意第三者在竊聽網路的時候擷取到這張影像時，只會從這張圖片去推斷通信的雙方是書畫的喜好者或文物的研究者，不會注意到這張圖片中含有隱藏的文字，資料便可秘密地傳送給了傳收方。

2. 研究方向與動機

偽裝術(steganography)的好壞取決於二種性質: 不可察覺性 (imperceptibility) 及 不可偵測性 (undetectability)。使用偽裝術是將秘密訊息嵌入掩護影像所產生的偽裝影像，應當讓肉眼難以察覺影像有所改變。嵌入或萃取的演算法都需要根據某些規則來修改像素，這些被修改過的像素應該避免被統計方法偵測出某種關連性，使得秘密訊息的存在被洩露出去。一旦秘密訊息的存在可以被第三人發現，偽裝術便失去作用，所以偽裝術不需要如同浮水印(watermark)一樣要求強韌性 (robustness)。

Least Significant Bit (LSB)取代法[7],[8]是一種快速且能大量藏入秘密訊息的方法,它將秘密訊息直接藏入所有像素值的最低位元 (Least Significant Bit) ，但是 LSB 本身卻存在容易被統計偵測的缺點(如: RS 偵測法[4-6])。為了利用 LSB 的方便,之後又有人提出了用 covering code 來改善 LSB 容易被統計偵測的缺點。Westfeld [10] 提出的 F5 是第一個實作出 matrix encoding 的方法。Matrix encoding [3],[7],[9] 能夠在 $2k$-1 個像素中藏入 k 個位元的秘密訊息並且使得 LSB 大大降低整張影像中的改變量。Zhang[11] 等人提出 “Hamming + 1” 的方法改進 matrix encoding 的藏量，在使用(7,4)Hamming code 的情況下，每 8 個像素可以藏入 4 個位元的秘密訊息。Chang [1,2]等人提出將 7 個像素當成一個區塊，利用 (7,4) Hamming code 重新分割(partition)產生的 covering code ，將秘密訊息以 covering code 進行編碼並藏入區塊中每個像素的最低位元。這個方法的藏量是每 7 個像素藏入 7 個位元的秘密訊息。

本篇論文主要改進 Chang [1],[2] 等人方法在偽裝影像品質方面的表現，提出 2 個低失真的資料隱藏方法：Scheme 1 可以將 3 個位元的秘密訊息藏入 4 個像素中而且最多只需改變一個像素而且被修改的像素是原來的像素 $+1$ 或 -1，所以影像品質會比較好。Scheme 2 利用 “± 1steganography”改善 Scheme 1 的藏量，由實驗結果可知，在相同的藏量下，Scheme 2 的 PSNR 及失真都比 Chang [1],[2]等人的方法好。

本篇論文主要是以減少失真率同時又保有高藏量為目標，第二章是相關文獻的討論，第三章描述本篇論文提出的二個方法，第四章是實驗數據及比較，第五章是結論及未來研究方向的討論。

二、相關文獻回顧

圖表等可以列在文中，或在參考文獻之後。列在文中者，請儘可能靠近正文中第一次提及的位置。比較大的圖表，可以跨兩個欄。各圖表請備說明內容，圖的說明請置於圖的下方，表的說明則請置於表的上方。

1. 漢明碼

設 V 是佈於 GF(2) 的 7 維向量空間且 C 為二元(7,4,3)線性漢明碼(binary Hamming linear code)，其中 $n = 2^3 - 1 = 7$ 是每個碼字(codeword)的位元個數，$k = 2^3 - 3 - 1 = 4$ 是碼字中訊息(information)的位元個數，還有 $r = n - k = 7 - 4 = 3$ 代表碼字中校驗位元(parity check bit)的位元個數，$\mathrm{d}(C) = 3$ 為 C 的碼距(code distance)。C 可視為 V 的一個 4 維向量子空間(vector subspace)，將 C 的任一組基底組成 4×7 矩陣 G 的列向量，此矩

陣G稱為C的生成矩陣(generator matrix)。G可將 4 維的訊息資料(information data)空間變換成V的 4 維向量子空間C。反之，V中任何 4 個線性獨立的向量可組成一個4×7的生成矩陣H來產生 1 個(7,4,3)的漢明碼。另一方面C可表示成為某個3×7矩陣H的零空間(null space of H)，矩陣H稱為C的校驗矩陣(parity-check matrix)，即$C=\{v\in V\mid vH^T=0\}$。生成矩陣G是用來編碼，而校驗矩陣H是用來解碼。G和H的關係可以相互轉換如下：

$$G=[I_k\mid P]_{k\times n}$$
$$H=[-P^T\mid I_{n-k}]_{(n-k)\times n}$$

因 d(C) = 3，所以C是一個 Error Correcting Code (ECC)可以糾錯 1 個位元，i.e.校驗矩陣H的充要條件是 rank $H=3$且任 2 個行向量是線性獨立。因校驗矩陣H的秩(rank)是 3，i.e. H的值域(range) $A=\{v\in V\mid vH^T\}$是 3 維的向量空間共有 8 個向量分別是$e_0=\begin{bmatrix}0\\0\\0\end{bmatrix}$, $e_1=\begin{bmatrix}0\\0\\1\end{bmatrix}$, ... , $e_7=\begin{bmatrix}1\\1\\1\end{bmatrix}$

很明顯地當收到 1 個 word $v\in V$時，經解碼器計算 symdrome $z=vH^T=0$，則是 codeword $v\in C$；若$z\neq0$，則$z\in\{e_1,...,e_7\}$，表示v不是 codeword，可以進行糾錯 1 位元的錯誤。

設$C_i=\{v\in V\mid vH^T=e_i\}$ $i=0,1,2,\ ...\ ,7$，則V可分割成 8 個 cosets $C_0,C_1,...,C_7$，$V=C_0\cup C_1\cup\ ...\cup C_7$，有些學者 [9-11]利用此特性來處理資料隱藏的問題。

2. Hamming + 1 Scheme

Zhang[11] 等人在 2007 年在 IEEE Communications letters 發表了 “Improving Embedding Efficiency of Covering Codes for Applications in Steganography”，提出了 Hamming + 1 的方法，可以直接改善 matrix encoding 的方法，讓原本失真很少的 matrix encoding 可以在不增加修改像素的情況下，增加 1 個位元的藏量。這個方法首先把掩護影像均分成許多區塊，假設每一個區塊有 m 個像素(pixles)，並且把這 m 個像素分成二部份：前面的 (m-1)個 pixels， 利用 matrix encoding[3,7,9]的方式在 lsb 內藏入秘密訊息；第二部份是指區塊中的最後一個 pixel(m-th)。把 m-th 的 pixel 的 lsb 和前面 (m-1) 個 pixels 的 second lsb 做 exclusive or 的運算，所得到的值 (0 或 1)去和秘密訊息做比較，藉此用來決定是否要修改這 m 個 pixels 值。

- 當 exclusive or 的值和秘密訊息相等時：

  (1)若第一部份中前面的 (m-1)個 pixels 它的 lsb 有被修改，表示這 (m-1) 個 pixels 用 matrix encoding 修改，最後第 m 個 pixel 不需要修改。

  (2)若第一部份中前面的 (m-1)個 pixels 它的 lsb 沒有被修改，這表示這個區塊中的所有 pixels 完全不需要修改。
- 當 exclusive or 的值和秘密訊息不相等時，則這 m 個 pixels 值，其修改規則如下：

  (1)若第一部份中前面的 (m-1)個 pixels 它的 lsb 有被修改，則任挑一被修改的 pixel 出來，並在這個 pixel 的 second lsb 做$0\leftrightarrow1$的動作，第 m-th 的 pixel 不需要修改。

  (2)若第一部份中前面的 (m-1)個 pixels 它的 lsb 沒有被修改，則我們僅需要修改第 m-th 的 pixel 值做$0\leftrightarrow1$的動作。

3. Chang 等人的方法

Chang[1] 等人在 2008 年在 International Symposium on Electronic Commerce and Security 發表了“A High Payload Steganographic Scheme Based on (7,4) Hamming Code for Digital Images”，這篇論文提出的方法先設 W 是佈於 GF(2) 的 7 維向量空間，$C=\{g_0^0,g_0^1,...,g_0^{15}\}$ 為 W 的 4 維子空間且C是一個 (7,4)漢明碼。將V

分割成 16 個類集分別為$G^0,G^1,\ldots,G^{15}$且每一個$G^u=\{g_0^u,g_1^u,\ldots,g_7^u\}$ $0\le u\le 15$,各自有 8 個 7 bit 的 word 且每一個 G^u 含一個漢明碼字 (Hamming codeword) g_0^u 。令H為一 (7,4) 漢明碼C的校驗矩陣，則$V=\{v=g_i^uH^T \mid i=0,1,2,\ldots,7\}$為一佈於 GF(2) 的 3 維空間，即$V=\left\{\begin{bmatrix}0\\0\\0\end{bmatrix},\ldots,\begin{bmatrix}1\\1\\1\end{bmatrix}\right\}$。當一個有 7 個像素的區塊$p_1,p_2,p_3,p_4,p_5,p_6,p_7$要藏入 7 個秘密訊息位元$s=s_1s_2s_3s_4s_5s_6s_7$。令$u=s_3s_5s_6s_7$及$v=s_1s_2s_4$。在 u-class 中，需求出某個$g_i^u$使得$v=g^uH^T$然後用 word g^u 取代$p_1,p_2,p_3,p_4,p_5,p_6,p_7$中的最低位元 (LSB)。取出時,假設拿到的偽裝影像的像素是$p_1',p_2',p_3',p_4',p_5',p_6',p_7'$，先取這些像素的 lsb 得到 7 位元的字串$l'=l_1'l_2'l_3'l_4'l_5'l_6'l_7'$，再取出 4 位元$u=l_3'l_5'l_6'l_7'$表示$l'$落在第$u$類別，計算$z=l'H^T$得到 symdrome 為 3 位元字串$z=z_1z_2z_3$，最後取出秘密訊息$s=z_1z_2l_3'z_3l_5'l_6'l_7'$。

現舉一例來說明，假設有 7 個像素$p=p_1,p_2,p_3,p_4,p_5,p_6,p_7$ =15,15,12,13,14,14,14 及 7 個位元的秘密訊息$s=s_1s_2s_3s_4s_5s_6s_7=1110011$。首先令$u=s_3s_5s_6s_7=1011$ 及 $v=s_1s_2s_4=110$。$G^{11}=\{g_0^{11}=(0110011),g_1^{11}=(1111001)$，$g_2^{11}=(0010011)$，$g_3^{11}=(1010011)$，$g_4^{11}=(0111011)$，$g_5^{11}=(1111011)$，$g_6^{11}=(0011011)$，$g_7^{11}=(1011011)\}$，存在 1 個 $g_v^u=g_6^{11}=(0011011)$ 使得$v=g_v^uH^T$。再以g_v^u取代 7 個像素p的最低位元得到偽裝影像p'=14,14,13,13,14,15,15。萃取秘密訊息時，先取出p'中所有像素的最低位元$l'=l_1'l_2'l_3'l_4'l_5'l_6'l_7'$ =(0 0 1 1 0 1 1)。計算$z=z_1z_2z_3=l'H^T=(1\,1\,0)$，最後得到秘密訊息$s=z_1z_2l_3'z_3l_5'l_6'l_7'$=(1 1 1 0 0 1 1)。

三、低失真的資料隱藏方法

1. (5,3) 線性碼

設$V=\{(a_1\,a_2\,a_3\,a_4\,a_5)\mid a_i\in \mathrm{GF}(2)\ i=1,2,3,4,5\}$ (GF(2)代表有 2 個元素的 Galois field)是佈於 GF(2) 的 5 維向量空間(vector space)。任何 1 個 3 維的向量子空間(vector subspace) C為(5,3)線性碼。設任意 1 個 rank 3 的矩陣$G=\begin{bmatrix}1&0&0&1&0\\0&1&0&0&1\\0&0&1&1&1\end{bmatrix}_{3\times 5}$為某 1 個(5,3)線性碼$C$的生成矩陣(generator matrix)。考慮用C將V分割(partition)成 4 個 cosets，其 leader vectors 分別為e_0=(0 0 0 0 0), e_1=(0 0 0 0 1), e_2=(0 0 0 1 0), e_3=(0 0 0 1 1)。令$e_i=(0\,0\,0\,e_\alpha\,e_\beta)$和$C_i=e_i+C$, $i=0,1,2,3$則

$$V=\bigcup_{i=0}^{3}C_i$$

表格 1 array decoding for (5,3) linear code

data	leader			
	$e_0 = 00000$	$e_1 = 00001$	$e_2 = 00010$	$e_2 = 00011$
(000)	00000	00001	00010	00011
(001)	00111	00110	00101	00100
(010)	01001	01000	01011	01010
(011)	01110	01111	01100	01101
(100)	10010	10011	10000	10001
(101)	10101	10100	10111	10110
(110)	11011	11010	11011	11000
(111)	11100	11101	11110	11111
coset	$C = C_0$	C_1	C_2	C_3

Remark 1.

設 $p_i = p_{i7}p_{i6}p_{i5}p_{i4}p_{i3}p_{i2}p_{i1}p_{i0}$ 為掩護影像的像素且 $0 < p_i < 255$。用 x 表示無關緊要的位元 (don't care bit)。現考慮 p_i 的最低 (least significant)2 位元 $p_{i1}p_{i0}$ 的 4 種 case：

1) 若 $p_{i1}p_{i0}$ = 00 或 10，則 $p_i' = p_i - 1$ 分別為 (xxxxxx11)及(xxxxxx01)。
2) 若 $p_{i1}p_{i0}$ = 01 或 11，則 $p_i' = p_i + 1$ 分別為 (xxxxxx10)及(xxxxxx00)。

所以 p_i 可選擇 +1 或 -1 使得 $p_i' = p_i - 1$ 或 $p_i' = p_i + 1$ 的值為 (xxxxxx$\overline{p_{i1}}\,\overline{p_{i0}}$)。(其中 $\overline{p_{i1}}$，$\overline{p_{i0}}$ 分別表示 p_{i1} 及 p_{i0} 的補數)

Remark 2.

任意 2-bit 的資料 d_1d_2，由表格 2 可知是對應到 2 個互補的 3-bit 字串 $w = c_1c_2c_3$ 及 $\overline{w} = \overline{c_1c_2c_3}$。因 2 個 3 位元字串的最大漢明距離(hamming distance)是 3，所以任意一個 3 位元字串 $a = a_1a_2a_3$，則 $\min(\mathrm{d}(w,a), \mathrm{d}(a,\overline{w})) \le 1$。

表格 2 code 與 資料 關係表

code $w = c_1c_2c_3$	data d_1d_2
000	00
001	11
010	01
011	10
100	10
101	01
110	11
111	00

2. Scheme 1: (將 3 位元的秘密訊息嵌入 4 個像素，最多只需更動 1 個像素值)

設 4 個像素組成一個區塊 $B = p_1, p_2, p_3, p_4$，$p_i = p_{i7}p_{i6}p_{i5}p_{i4}p_{i3}p_{i2}p_{i1}p_{i0}$，$i = 1,2,3,4$，3 位元的秘密訊息 $s = s_1s_2s_3$。

□ 嵌入方法

首先取 s_1s_2 查表格 2，得到對應 2 個 3-bit 字串 $c_1c_2c_3$ 及 $\overline{c_1c_2c_3}$。由 Remark 2 可知 $c_1c_2c_3$ 及 $\overline{c_1c_2c_3}$ 中有 1 個字串與 $p_{10}p_{20}p_{30}$ 的漢明距離小於等於 1。假設 $\mathrm{d}(c_1c_2c_3, p_{10}p_{20}p_{30}) \le 1$，現分 2 個 cases 討論嵌入情況：

case 1 $\mathrm{d}(c_1c_2c_3, p_{10}p_{20}p_{30}) = 0$

(a) 若 $p_{11} \oplus p_{21} \oplus p_{31} \oplus p_{40} = s_3$，則無需更改 B 中的任何 1 個像素。

(b) 若 $p_{11} \oplus p_{21} \oplus p_{31} \oplus p_{40} \neq s_3$，則用 $\overline{p_{40}}$ 取代 p_{40}。

所以更動 p_4 像素值為 p_4' 的機率為 $\frac{1}{8} \times \frac{1}{2}$。

case 2 $\mathrm{d}(c_1c_2c_3, p_{10}p_{20}p_{30}) = 1$ 表示 $c_1c_2c_3$ 與 $p_{10}p_{20}p_{30}$ 中有 1 個位元是相異。假設 $p_{10} \neq c_1$。

(a) 若 $p_{11} \oplus p_{21} \oplus p_{31} \oplus p_{40} = s_3$，則用 $\overline{p_{10}}$ 取代 p_{10}。

(b) 若 $p_{11} \oplus p_{21} \oplus p_{31} \oplus p_{40} \neq s_3$。則

$p_1^{'}=p_1+1$ 或 $p_1^{'}=p_1-1$ 使得 $p_1^{'}=(\text{xxxxxx}\overline{p_{11}}\,p_{10})$。

(c) 其他 case 如 $p_{20}\neq c_2$ 及 $p_{30}\neq c_3$ 處理情形同(a)(b)方法。

所以 case2 更動 1 個像素值的機率為 $\frac{7}{8}\times\frac{1}{2}\times 2$。

□ 萃取方法

設偽裝影像的 1 個區塊 $B'=p_1',p_2',p_3',p_4'$ 且 $p_i'=p_{i7}'p_{i6}'p_{i5}'p_{i4}'p_{i3}'p_{i2}'p_{i1}'p_{i0}'$，$i=1,2,3,4$，首先取 $p_{10}'p_{20}'p_{30}'$ 經由表格 2 得到秘密訊息的前 2 個位元 s_1s_2。再計算 $p_{11}'\oplus p_{21}'\oplus p_{31}'\oplus p_{40}'=s_3$ 得到秘密訊息的第 3 個位元 s_3。所以可以從 4 個像素的區塊 B 取出 3 個位元的秘密訊息 $s=s_1s_2s_3$。

□ 失真率

由上述嵌入的方式可明顯看出：

- 更動 0 個像素機率是 $\frac{1}{8}\times\frac{1}{2}$
- 更動 1 個像素機率是 $\frac{1}{8}\times\frac{1}{2}+\frac{7}{8}\times\frac{1}{2}\times 2$

1) 平均每個區塊中被更改的像素個數為 $0\times\frac{1}{8}\times\frac{1}{2}+\left[\frac{1}{8}\times\frac{1}{2}+\frac{7}{8}\times\frac{1}{2}\times 2\right]=\frac{1}{1}$，

2) 失真率(每個像素平均更動的機率) $\frac{15}{16}\times\frac{1}{4}=\frac{15}{64}\approx 0.2344$。

3) 平均 PSNR ≈ 54.4726。

例 3.1

如下圖，有一個 4 個像素的區塊 $B=p_1,p_2,p_3,p_4$=12,15,100,127 及欲藏入秘密訊息 s=110。先取 $s_1s_2=11$ 查 Table 2 得到二組候選碼 001 及 110，分別計算這二組候選碼與 p_1,p_2,p_3 的最低位元 010 之間的漢明距離，選擇漢明距離小於等於 1 的那一組。最後會選擇 110，是因為 010 與 110 只有一個位元不同，漢明距離等於 1；如果是選 010 則與 001 會有二個位元不同，漢明距離等於 2。因此需要修改像素 p_1 同時檢查 s_3 是否等於 $p_{11}\oplus p_{21}\oplus p_{31}\oplus p_{40}$。這個例子中 s_3=0, p_{11}=0, p_{21}=1, p_{31}=0, p_{40}=1，所以結果是 $s_3=0=p_{11}\oplus p_{21}\oplus p_{31}\oplus p_{40}$，不需要再修改 p_{11}。故用 $\overline{p_{10}}=1$ 取代 $p_{10}=0$，最後得到新的像素值 $p_1^{'}=13$。這個區塊的偽裝影像像素值為 13,15,100,127。

p1	12	0	0	0	0	1	1	0	0
p2	15	0	0	0	0	1	1	1	1
p3	100	0	1	1	0	0	1	0	0
p4	127	0	1	1	1	1	1	1	1

Figure 1 例 3.1 之掩護影像中像素與位元關係圖

p'1	13	0	0	0	0	1	1	0	1
p'2	15	0	0	0	0	1	1	1	1
p'3	100	0	1	1	0	0	1	0	0
p'4	127	0	1	1	1	1	1	1	1

Figure 2 例 3.1 之偽裝影像中像素與位元關係圖

考慮取出的情況，從偽裝影像中取出一個區塊 $B'=p_1^{'},p_2^{'},p_3^{'},p_4^{'}$=13,15,100,127，從 $p_1^{'},p_2^{'},p_3^{'}$ 取出 $p_{10}^{'}p_{20}^{'}p_{30}^{'}$=110，以 $p_{10}^{'}p_{20}^{'}p_{30}^{'}$ 查表格 2 可以知道原來的秘密訊息 s_1s_2= 11。計算秘密訊息的第三個位元 $s_3=p_{11}^{'}\oplus p_{21}^{'}\oplus p_{31}^{'}\oplus p_{40}^{'}=0\oplus 1\oplus 0\oplus 1=0$，最後可知秘密訊息為 $s=s_1s_2s_3=110$。

3. Scheme 2:(將 6 位元的秘密訊息嵌入 6 個像素的區塊，最多更動 3 個像素。)

設 6 個像素組成一個區塊 $B=p_1,p_2,p_3,p_4,p_5,p_6$，$p_i=p_{i7}p_{i6}p_{i5}p_{i4}p_{i3}p_{i2}p_{i1}p_{i0}$，$i=1,2,3,4,5,6$，以及準備嵌入的秘密訊息 $s=s_1s_2s_3s_4s_5s_6$。首先視 v

為V中的一個向量，因此v會落在其中一個coset $C_i = e_i + C$內，$\therefore v - e_i = (s_1 \quad s_2 \quad s_3 \quad s_4 - e_{i0} \quad s_5 - e_{i1}) \in C$，故存在一個data向量$d = (d_1 \quad d_2 \quad d_3)$使得$dG = v - e_i$。

□ 嵌入方法

討論嵌入的秘密訊息$s = s_1 s_2 s_3 s_4 s_5 s_6$的各種情況。

case 1 $d(p_{10}p_{20}p_{30}, d_1d_2d_3) = 0$且

1) $\begin{cases} p_{11} \oplus p_{40} = e_{i0} & (1) \\ p_{21} \oplus p_{50} = e_{i1} & (2) \\ p_{31} \oplus p_{60} = s_6 & (3) \end{cases}$則修改0個像素，

其機率是$\binom{3}{0}\frac{1}{8} \times \frac{1}{8}$。

2) 若(1), (2), (3)式中有1個式子不成立，假設$p_{11} \oplus p_{40} \neq e_{i0}$，則修改1個像素$p_{40}$，以$\overline{p_{40}}$取代$p_{40}$；修改1個像素的機率是$\binom{3}{0}\frac{1}{8} \times \binom{3}{2}\frac{2}{8}$。

3) 若(1), (2), (3)式中有2個式子同時不成立，假設$p_{11} \oplus p_{40} \neq e_{i0}$及$p_{21} \oplus p_{50} \neq e_{i1}$，則修改2個像素$p_{40}$和$p_{50}$，用$\overline{p_{40}}$與$\overline{p_{50}}$去取代$p_{40}$及$p_{50}$；修改2個像素的機率是$\binom{3}{0}\frac{1}{8} \times \binom{3}{2}\frac{1}{8}$。

4) 若(1), (2), (3)式同時不成立，則修改3個像素p_{40}, p_{50}, p_{60}，用$\overline{p_{40}}, \overline{p_{50}}, \overline{p_{60}}$去取代$p_{40}, p_{50}$及$p_{60}$；修改3個像素的機率是$\binom{3}{0}\frac{1}{8} \times \binom{3}{3}\frac{1}{8}$。

case 2 $d(p_{10}p_{20}p_{30}, d_1d_2d_3) = 1$，表示$p_{10}p_{20}p_{30}$與$d_1d_2d_3$恰有1個位元相異，假設是$p_{10} \neq d_1$。

1) 修改1個像素的情況：

a) 若(1), (2), (3)式皆成立，則修改p_1的像素值，用$\overline{p_{10}}$取代p_{10}。

b) 若(1)式不成立，則修改p_1成$p_1' = p_1 + 1$或$p_1 - 1$使得$p_{11}p_{10}$成$\overline{p_{11}p_{10}}$(見Remark 1)。

所以修改1個像素的機率是$\binom{3}{1}\frac{1}{8} \times \frac{1}{8} + \binom{3}{1}\frac{1}{8} \times \frac{1}{8}$。

2) 修改2個像素的情況：若(1),(2)式都不成立，則將p_1修改成$p_1' = p_1 + 1$或$p_1 - 1$使得$p_{11}p_{10}$變成$\overline{p_{11}}\,\overline{p_{10}}$；將$p_{50}$換成$\overline{p_{50}}$。相同方式處理(1), (3)式不成立的情況，所以修改2個像素的情況發生的機率是$\binom{3}{1}\frac{1}{8} \times \binom{2}{1}\frac{1}{8}$。

3) 修改3個像素的情況：

a) 若(2), (3)式都不成立，則將$\overline{p_{10}}, \overline{p_{50}}$及$\overline{p_{60}}$取代$p_{10}, p_{50}$及$p_{60}$。

b) 若(1),(2),(3)式都不成立，則將$\overline{p_{50}}, \overline{p_{60}}$取代$p_{50}$及$p_{60}$。用$p_1' = p_1 + 1$或$p_1 - 1$取代$p_1$使得$p_{11}p_{10}$變成$\overline{p_{11}}\,\overline{p_{10}}$。

所以修改3個像素的機率為$\binom{3}{1}\frac{1}{8} \times \frac{1}{8} \times 2$。

case 3 $d(p_{10}p_{20}p_{30}, d_1d_2d_3) = 2$，$p_{10}p_{20}p_{30}$與$d_1d_2d_3$恰有2個位元相異，假設是$p_{10}p_{20} \neq d_1d_2$。

1) 修改2個像素的情況：

a) 若(1),(2),(3)式都成立，則用 $\overline{p_{10}}, \overline{p_{20}}$ 取代 p_{10}, p_{20}。

b) 若(1)式不成立， $p_{11} \oplus p_{40} \neq e_{i0}$ 則用 $\overline{p_{20}}$ 取代 p_{20}，以 $p_1' = p_1 + 1$ 或 $p_1 - 1$ 取代 p_1 使得 $p_{11}p_{10}$ 成 $\overline{p_{11}}\,\overline{p_{10}}$。若(2), (3)式不成立以相同的方式處理之。

c) 若 (1), (2) 式都不成立，則用 $p_i' = p_i + 1$ 或 $p_i' = p_i - 1$ 使得 $p_{i1}p_{i0}$ 變成 $\overline{p_{i1}}\,\overline{p_{i0}}$ $i = 1,2$。

所以修改 2 個像素的機率是 $\binom{3}{2}\frac{1}{8}\times\frac{1}{8}\times 2^2$。

2) 修改 3 個像素的情況：

a) 若(3)式不成立，則用 $\overline{p_{10}}, \overline{p_{20}}$ 及 $\overline{p_{60}}$ 取代 p_{10}, p_{20} 及 p_{60}。

b) 若(1),(3)式不成立，則用 $p_1' = p_1 + 1$ 或 $p_1' = p_1 - 1$ 使得 $p_{11}p_{10}$ 變成 $\overline{p_{11}}\,\overline{p_{10}}$，並用 $\overline{p_{20}}\,\overline{p_{50}}$ 取代 p_{20} 及 p_{50}。若(2), (3)式不成立，亦如同(1),(3)式不成立的方式處理。

c) 若 (1),(2),(3) 式皆不成立，則用 $p_1' = p_1 + 1$ 或 $p_1 - 1$ 及 $p_2' = p_2 + 1$ 或 $p_2 - 1$ 分別取代 p_1, p_2 及用 $\overline{p_{60}}$ 取代 p_{60}。

所以修改 3 個像素的情況發生的機率是 $\binom{3}{2}\frac{1}{8}\times\frac{1}{8}\times 2^2$。

case 4 $d(p_{10}p_{20}p_{30}, d_1d_2d_3) = 3$，表示 $\overline{p_{10}} = d_1$, $\overline{p_{20}} = d_2$, $\overline{p_{30}} = d_3$。修改 3 個像素的情況有：

a) 若 (1),(2),(3) 式都成立，則用 $\overline{p_{10}}, \overline{p_{20}}, \overline{p_{30}}$ 取代 p_{10}, p_{20}, p_{30}。

b) 若 (1),(2),(3) 其中有 1 個式子不成立，假設 (1) 式不成立，則用 $p_1' = p_1 + 1$ 或 $p_1 - 1$ 取代 p_1。用 $\overline{p_{20}}, \overline{p_{30}}$ 分別取代 p_{20}, p_{30}。其他情況 (2) 或 (3) 式不成立亦用相同方式處理。

c) 若(1), (2),(3)其中有 2 個式子不成立，例如(1), (2)式不成立，則用 $p_1' = p_1 + 1$ 或 $p_1 - 1$ 及 $p_2' = p_2 + 1$ 或 $p_2 - 1$ 取代 p_1 及 p_2。其他 case 亦以同樣的方式處理。

d) 若 (1),(2),(3) 皆不成立，則用 $p_1' = p_1 + 1$ 或 $p_1 - 1$ 及 $p_2' = p_2 + 1$ 或 $p_2 - 1$ 及 $p_3' = p_3 + 1$ 或 $p_3 - 1$ 取代 p_1, p_2 及 p_3。

所以修改 3 個像素發生的機率是 $\binom{3}{3}\frac{1}{8}\times\left[\binom{3}{0}+\binom{3}{1}+\binom{3}{2}+\binom{3}{3}\right]=\binom{3}{3}\frac{1}{8}\times\frac{1}{8}\times 2^3$

□ 萃取方法

將 6 個像素組成一個區塊 $B' = p_1', p_2', p_3', p_4', p_5', p_6'$，$p_i' = p_{i7}'p_{i6}'p_{i5}'p_{i4}'p_{i3}'p_{i2}'p_{i1}'p_{i0}'$ $i = 1,2,3,4,5,6$。首先取得 $d = p_{10}'p_{20}'p_{30}'$，$e_\alpha = p_{11}' \oplus p_{40}'$，$e_\beta = p_{20}' \oplus p_{50}'$，$s_6 = p_{31}' \oplus p_{60}'$，計算 $dG \oplus (000e_\alpha e_\beta) = s_1s_2s_3s_4s_5$ 得到秘密訊息的前 5 個位元。再串接 s_6 得到秘密訊息

$s = s_1 s_2 s_3 s_4 s_5 s_6$。所以可以從 6 個像素的區塊中取出 6 個位元的秘密訊息。

□ 失真率

Scheme 2 可以將 6 個位元的秘密訊息 $s = s_1 s_2 s_3 s_4 s_5 s_6$，嵌入 6 個像素的區塊 $B = p_1, p_2, p_3, p_4, p_5, p_6$，最多更改 3 個像素。

- 平均每個區塊中被更改的像素個數為：

$$0\times\binom{3}{3}\frac{1}{8}\times\frac{1}{8}+1\times\left[\binom{3}{0}\frac{1}{8}\times\frac{1}{8}\times 3+\binom{3}{1}\frac{1}{8}\times\frac{1}{8}\times 2\right]+$$

$$2\times\left[\binom{3}{0}\frac{1}{8}\times\frac{1}{8}\times 3+\binom{3}{1}\frac{1}{8}\times\frac{1}{8}\times 2+\binom{3}{2}\frac{1}{8}\times\frac{1}{8}\times 2^2\right]+$$

$$3\times\left[\binom{3}{0}\frac{1}{8}\times\frac{1}{8}\times 1+\binom{3}{1}\frac{1}{8}\times\frac{1}{8}\times 2+\binom{3}{2}\frac{1}{8}\times\frac{1}{8}\times 4+\binom{3}{3}\frac{1}{8}\times\frac{1}{8}\times 2^3\right]$$

$$=\frac{9}{64}+\frac{2\times 21}{64}+\frac{3\times 27}{64}=\frac{132}{64}=\frac{33}{16}$$，所以每 1 個像素平均失真為 $\frac{33}{16}\times\frac{1}{6}=\frac{33}{96}\approx 0.3438$。

- 平均 PSNR ≈ 52.7684。

例 3.2

若掩護影像有一個 6 個像素的區塊 $B = p_1, p_2, p_3, p_4, p_5, p_6 = 100,113,111,97,102,107$ 及 6 個位元的秘密訊息 $s = s_1 s_2 s_3 s_4 s_5 s_6 =$ 101001。首先查表格 1 可知 $s_1 s_2 s_3 s_4 s_5 = 10100 = (101)G \oplus (00001) = dG \oplus e_1$。因為 $p_{10} \neq d_1$ 且 $p_{11} \oplus p_{40} \neq e_{11}$，所以將 p_1 以 -1 進行修改，修改完的像素 $p_1' = p_1 - 1 = 99$。接著因為 $p_{20} \neq d_2$ 且 $p_{21} \oplus p_{50} \neq e_{10}$，所以將 p_2 以 +1 進行修改，修改完的像素 $p_2' = p_2 + 1 = 114$。最後因為 $p_{30} = d_3$ 且 $p_{31} \oplus p_{60} \neq s_6$，所以將 p_6 以 -1 進行修改，修改完的像素 $p_6' = p_6 - 1 = 106$。得到一個 6 個像素的偽裝影像的區塊 $B' = 99,114,111,97,102,106$。

p_1	011001	0	0
p_2	011100	0	1
p_3	011011	1	1
p_4	011000	0	1
p_5	011001	1	0
p_6	011010	1	1

Figure 3 例 3.2 之掩護影像中像素與位元關係圖

p_1	011000	1	1
p_2	011100	1	0
p_3	011011	1	1
p_4	011000	0	1
p_5	011001	1	0
p_6	011010	1	0

Figure 4 例 3.2 之偽裝影像中像素與位元關係圖

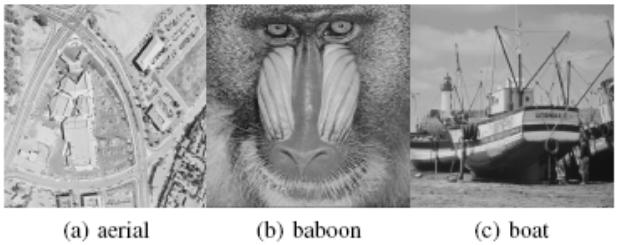
從偽裝影像取得一個 6 個像素的區塊 $B' = p_1', p_2', p_3', p_4', p_5', p_6' = 99,114,111,97,102,106$，欲從其中取出 6 個位元的秘密訊息。先從 p_1', p_2', p_3' 中取出 data $d = (p_{10}' \quad p_{20}' \quad p_{30}') = (1\ 0\ 1)$。計算 $e_\alpha = p_{11}' \oplus p_{40}' = 0$，$e_\beta = p_{21}' \oplus p_{50}' = 1$ 及 $s_6 = p_{31}' \oplus p_{60}' = 1$。因為 data $d = (1\ 0\ 1)$ 及 $e_i = (0\,0\,0\,e_\alpha\ e_\beta) = (0\ 0\ 0\ 0\ 1)$，計算 $dG \oplus e_i = (10100) = s_1 s_2 s_3 s_4 s_5$ 還原秘密訊息的前 5 個位元，最後串接上 s_6 便得到原來的秘密訊息 $s =$ 101001。

四、實驗與討論

實驗在 AMD Athlon64 X2 3800+及 1GB 記憶體的 PC 上使用 gcc, g++ 在 Ubuntu 8.04 的作業系統下進行開發測試。並且使用以下 7 張圖片做為掩護影像來進行實驗。

Table 3 Experiment Platform

CPU	AMD Athlon64 X2 3800+
Memory	1GB
Operating System	Ubuntu Release 8.04 Kernel 2.6.24-24-generic
Development Environment	gcc, g++



(a) aerial (b) baboon (c) boat

(d) frog (e) goldhill (f) lena

(g) peppers

Figure 5 實驗中用到的掩護影像

Table 4 是表示因為 4 個方法對像素只有做 ± 1 的修改，所以可以分別計算 Hamming + 1, Scheme 1, Scheme 2, Chang et. al 這 4 個方法理論上的藏入的秘密訊息數與區塊中的像素數的比值,平均 distortion 及平均 PSNR。

Table 4 Scheme 1, Hamming + 1, Scheme 2, Chang et. al 的比較表

	Hamming +1	Scheme 1	Scheme 2	Chang *et. al*
Bit per pixel (bpp)	4/8 = 0.50	3/4 = 0.75	6/6 = 1.00	7/7 = 1.00
Average distortion	0.1250	0.2344	0.3438	0.5000
Average PSNR	57.1617	54.4726	52.7677	51.1411

Table 5~6 是把一組隨機的 01 字串當成秘密訊息同時放入不同的七個掩護影像內並比較其 PSNR 值及 distortion。

Table 5 Hamming + 1 與 Scheme 1 的 PSNR 與 distortion 比較表

cover image	Hamming + 1		scheme 1	
	PSNR	distortion	PSNR	distortion
aerial.bmp	57.4024	0.1173	54.6990	0.2184
baboon.bmp	57.4417	0.1172	54.7298	0.2187
boat.bmp	57.4363	0.1173	54.7164	0.2195
frog512.bmp	57.2861	0.1173	54.5491	0.2196
goldhill.bmp	57.4308	0.1175	54.7291	0.2189
lena.bmp	57.4445	0.1171	54.7279	0.2189
peppers.bmp	57.4387	0.1173	54.7347	0.2185

Table 6 Scheme 2 與 Chang et. al 的 PSNR 與 distortion 比較表

cover image	scheme 2		Chang et.al	
	PSNR	distortion	PSNR	distortion
aerial.bmp	52.3601	0.3756	51.1381	0.5003
baboon.bmp	52.3982	0.3741	51.1474	0.4993
boat.bmp	52.3965	0.3745	51.1192	0.5025
frog512.bmp	52.3035	0.3722	51.1428	0.4998
goldhill.bmp	52.3877	0.3752	51.1483	0.4992
lena.bmp	52.3945	0.3747	51.1398	0.5002
peppers.bmp	52.3761	0.3762	51.1401	0.5001

Table 7~10 是取不同影像的像素的最高位元 (Most Significant Bit) 當做秘密訊息藏入相同的掩護影像，藉此實驗本篇的方法嵌入不同的秘密訊息時，是否對特定種類的圖產生較多的失真。首先是不同的秘密訊息嵌入 lena 的實驗結果。

Table 7 用 Hamming + 1 與 Scheme 1 法將七種不同的秘密訊息藏入 lena 圖中的 PSNR 與 distortion 比較表

secret source	Hamming + 1		scheme 1	
	PSNR	distortion	PSNR	distortion
aerial	57.4536	0.1169	54.7524	0.2177
boat	57.4450	0.1171	54.7344	0.2186
baboon	57.4424	0.1172	54.7458	0.2180
lena	57.4404	0.1172	54.7454	0.2180
frog512	57.4579	0.1168	54.7447	0.2181
peppers	57.4409	0.1172	54.7496	0.2178
goldhill	57.4389	0.1173	54.7479	0.2179

Table 8 用 Scheme 2 與 Chang et.al 法將七種不同的秘密訊息藏入 lena 圖中的 PSNR 與 distortion 比較表

secret source	scheme 2		Chang et.al	
	PSNR	distortion	PSNR	distortion
aerial	52.3920	0.3749	51.1444	0.4996
boat	52.3985	0.3743	51.1464	0.4994
baboon	52.3936	0.3747	51.1386	0.5003
lena	52.3877	0.3752	51.1147	0.5030
frog512	52.3869	0.3753	51.1282	0.5015
peppers	52.3806	0.3759	51.1391	0.5002
goldhill	52.3938	0.3747	51.1306	0.5012

接著比較不同的秘密訊息嵌入 baboon 的實驗結果。

Table 9 用 Hamming + 1 與 Scheme 1 將七種不同的秘密訊息藏入 baboon 圖中的 PSNR 與 distortion 比較表

secret source	Hamming + 1		scheme 1	
	PSNR	distortion	PSNR	distortion
aerial	57.4421	0.1171	54.7378	0.2183
boat	57.4418	0.1171	54.7388	0.2182
baboon	57.4352	0.1174	54.7214	0.2192
lena	57.4509	0.1169	54.7320	0.2186
frog512	57.4489	0.1170	54.7188	0.2193
peppers	57.4458	0.1170	54.7397	0.2182
goldhill	57.4444	0.1171	54.7292	0.2189

Table 10 用 Scheme 2 與 Chang et. al 法將七種不同的秘密訊息藏入 baboon 圖中的 PSNR 與 distortion 比較表

secret source	scheme 2		Chang et.al	
	PSNR	distortion	PSNR	distortion
aerial	52.4056	0.3736	51.1532	0.4986
boat	52.3932	0.3747	51.1442	0.4996
baboon	52.3878	0.3752	51.1041	0.5043
lena	52.3923	0.3748	51.1443	0.4996
frog512	52.3818	0.3757	51.1296	0.5013
peppers	52.3914	0.3748	51.1382	0.5003
goldhill	52.3969	0.3744	51.1404	0.5001

由 Table 5~10 的實驗結果可發現，Scheme 1 與 Hamming + 1 的方法相比，Scheme 1 的 distortion 與 PSNR 較差，但藏量多了 1/4 bpp。Scheme 2 的方法與 Chang 等人相比，Scheme 2 在藏量上與 Chang 等人的方法相同時， Scheme 2 在 distortion 方面較 Chang 等人的方法改善 0.13。

五、結論與未來研究方向

對於灰階數位影像的資料隱藏，本文提出二個 schemes。這二個 schemes 皆利用 (5,3)線性碼的特性同時達成低失真及高藏量的目標。Scheme 1 平均每個像素可藏 $\frac{3}{4}$ 位元的秘密訊息，平均失真率為 0.2344，平均 PSNR 為 54.4726。Scheme 2 平均每個像素可藏 1 位元的秘密訊息，平均失真率為 0.3438，平均 PSNR 為 52.7677。今後希望能夠探討使用編碼將秘密訊息藏入黑白數位影像的資料隱藏方法是否可以得到一些更好的結果。

六、參考文獻

[1] C.-C. Chang, T. D. Kieu, and Y.-C. Chou. “A high payload steganographic scheme based on (7,4) hamming code for digital images”, In International Symposium on Electronic Commerce and Security, pages 16–21, 2008.

[2] C.-C. Chang, T. D. Kieu, and Y.-C. Chou. Using nearest coveringcodes to embed secret

information in grayscale images. In ICUIMC ’08: Proceedings of the 2nd international conference on Ubiquitous information management and communication, pages 315–320. ACM, 2008.

[3] R. Crandall. Some notes on steganograph. http://os.inf.tudresden.de/ westfeld/crandall.pdf, 1998.

[4] J. Fridrich and M. Goljan. Practical steganalysis of digital images - state of the art. In In Proceedings of SPIE, pages 1–13, 2002.

[5] J. Fridrich, M. Goljan, and R. Du. Detecting lsb steganography in color and gray-scale images. IEEE MultiMedia, 8(4):22–28, 2001.

[6] J. Fridrich, M. Goljan, and R. Du. Reliable detection of lsb steganography in color and grayscale images. IEEE Multimedia, 8:22–28, 2001.

[7] M. Khatirinejad and P. Lisonˇek. Linear codes for high payload steganography. Discrete Applied Mathematics, 157:971–981, March 2009.

[8] J. Mielikainen. Lsb matching revisited. Signal Processing Letters, 13:285–287, May May 2006.

[9] C. K. Santosh Arjun, Atul Negi and D. Keerthi. An approach to adaptive steganography based on. In TENCON 2007 – 2007 IEEE Region 10 Conference, pages 1–4, 2007.

[10] A. Westfeld. F5-a steganographic algorithm. In IHW ’01: Proceedings of the 4th International Workshop on Information Hiding, pages 289–302. Springer-Verlag, 2001.

[11] W. Zhang, S. Wang, and Z. Xinpeng. Improving embedding efficiency of covering codes for applications in steganography. In Communications Letters, IEEE, volume 11, pages 680–682, 2007.