

一個改良的利用兩個像素之差異級別的影像偽裝術¹

陳玉璽
逢甲大學
通訊工程系

谷秋月
逢甲大學
電機與通訊工程所
南開技術學院

林秀峰
逢甲大學
資訊工程系

陳志滢
逢甲大學
通訊工程系

fcueevc@yahoo.com.tw

moon384@nkc.edu.tw

hfin@fcu.edu.tw

chihchen@fcu.edu.tw

摘要

2003 年, Wu 與 Tsai 提出一個根據相鄰的兩個像素值的差異級別來嵌入不同數量的訊息位元的灰階影像偽裝術, 同時根據實驗結果指出他們的方法不但具有良好的不可察覺性而且可以抵擋由 Fridrich 等人在 2001 年所發表的 RS 統計偵測的攻擊。

由於一個真正實用有效的影像偽裝術, 除了要具備難以察覺與難以偵測的安全性外, 尚須要求具有盡可能高的嵌入容量; 本論文參考 Wu 與 Tsai 的作法提出一個改良的嵌入技術。接著, 我們進行一系列的實驗。實驗結果證明, 本論文的方法與 Wu 與 Tsai 的方法一樣的, 也具有相當良好的不可察覺性並且可以抵擋 Fridrich 等人的 RS 統計偵測攻擊。最後, 實驗的結果進一步顯示, 在安全的前提下, 本論文的作法可以比 Wu 與 Tsai 的方法嵌入更多的訊息位元。因此, 我們的方法更適於偽裝術的實際應用。本研究接受國科會計畫編號 NSC95-2221-E-035-047 之部分經費補助

關鍵詞：影像偽裝術、偽裝分析、RS 偽裝分析法、空間域最低位元嵌入法、區塊平滑度。

Abstract

In 2003 Wu and Tsai proposed a steganographic method for images by pixel-value differencing. Experiments results show that their method not only possesses high imperceptibility but also is capable of resisting the RS stegoanalysis suggested by Fridrich *et al* in 2001.

¹ 本研究接受國科會計畫編號 NSC 95- 2221 -E-035-047 之部分經費補助

Owing to the fact that in addition to high imperceptibility and undetectability, a practical image steganography also requires as high embedding capacity as possible. In this paper, we propose an improved embedding scheme by taking advantage of Wu and Tsai's method.

A series of experiments show that our scheme, as well as Wu and Tsai's scheme, possesses high imperceptibility and is capable of resisting the RS stegoanalysis. Finally, experiment results also show that our approach is able to embed more data than Wu and Tsai's method. Therefore, our method is fairly suitable for practical application of steganography.

Keywords: Steganography 、Stegoanalysis、RS Steganalysis、Spatial LSB embedding、Block smoothness.

1. 前言

資訊偽裝學 (Steganography), 簡稱為偽裝學, 是近年來在國際上引發極度關注與興趣的研究主題。它是資訊隱藏學 (Information Hiding) 的一個分支, 也是一種有別於密碼學的加/解密作法的秘密通信技術, 主要是藉由隱藏秘密訊息的存在性來獲得秘密通信的安全[1-2, 4, 11]。據報導, 在 911 事件中, 恐怖組織可能就是利用偽裝技術做為傳遞攻擊指令的工具; 還有一些電腦駭客也可能利用偽裝技術做為電腦病毒寄生與散播的管道。另一方面, 聯合國的人權組織則大力鼓勵一些受暴政統治的國家人民利用安全的偽裝技術突破統治者的層層封鎖來揭發暴力統治的事實。因此, 偽裝學可說是攸關國家社會安全與維護人權的一個非常重要的研究課題。

數位影像偽裝術(簡稱影像偽裝術; Image Steganography)是利用非機密性的數位影像來

掩護及傳送可以用位元串 (bit stream) 表示的秘密訊息,使得發送與接收雙方之外的第三者 (以下稱為攻擊者)難以察覺出或偵測出秘密訊息的存在以及通訊行為之進行的一種技術。其中,用來掩護秘密訊息的影像稱為掩護影像 (cover image)。掩護的方法,稱為訊息嵌入法 (embedding method),通常是將秘密訊息的位元嵌入掩護影像在空間域的像素 (稱為空間域嵌入法) 或頻率域的係數 (稱為頻率域嵌入法)。訊息嵌入後,得到一個與原始掩護影像相近似的影像稱為偽裝影像 (stego image)。而由偽裝影像中取出嵌入訊息的作法,則稱為訊息萃取法 (extracting method) [1-2, 4, 11]。

影像偽裝術的安全性取決於不可察覺 (imperceptibility) 以及不可偵測 (undetectability) 秘密訊息之存在的程度。所以,影像偽裝術的重點在於能否設計一套有效的訊息嵌入/萃取法使得攻擊者很難以肉眼察覺出或以統計方法偵測出偽裝影像中秘密訊息的存在。也因為一旦被偵測出秘密訊息的存在,便算失敗;所以,一般的影像偽裝術不必如浮水印 (watermarking) 般要求具有高度的強韌性 (robustness)。然而,為了提高偽裝影像對自然雜訊的容忍度,也可以在安全性的前提下賦予適度的強韌性。此外,在安全性的前提下,一個實用有效的影像偽裝術,尚須要求具有盡可能高的嵌入容量 (embedding capacity) 以及盡可能低的計算複雜度 (computation complexity)。

空間域嵌入法具有觀念簡單、計算容易以及嵌入容量較大的優點。其中,最典型的作法乃是將掩護影像在空間域的某些連續或隨機選取的像素的 LSB (Least Significant Bit) 位元由秘密訊息位元來取代,故稱為空間域 LSB 嵌入法,簡稱為 LSB 嵌入法 [3, 5-7, 9-10, 12, 14-16]。許多常用的偽裝術軟體如 EZstego、Hide&seek、S-Tool4、Steganos 以及 Stego Dos 等都是採用 LSB 嵌入的作法。

雖然,LSB 法通常具有良好的不可察覺性與較高的嵌入容量等優點;然而,也常會因選用的掩護影像不當、選取的像素集合不當、或使用的嵌入作法不當而改變自然影像的某些統計特性。因此,利用適當的統計量加以檢驗與分析,常可偵測出肉眼無法察覺的異常現象 [13]。這種統計檢驗與分析的技術稱為偽裝分析學 (Steganalysis)。它一方面可以嚇阻不法人士濫用偽裝術做為犯罪工具,另一方面也可

以積極的促進和提高偽裝演算法的安全使之能夠應用在例如揭發暴政與保護人權等正面用途。因此,也成為目前及未來偽裝學的一個非常重要的研究方向。第一個針對 LSB 嵌入法的偽裝分析法是由 Westfeld 與 Pfizmann [14] 於 1999 年利用像素值的卡方分配 (Chi-square distribution) 統計量提出的。接著, Fridrich 等人 [7, 8] 於 2001 年利用不同平滑度的像素群之分佈特性, Avcibas 等人 [3] 於 2001 年利用影像品質度量之統計特性, 以及 Farid [6] 於 2002 年利用小波分解的高階統計量, 分別提出不同的 LSB 偽裝分析法。其中, Avcibas 等人 [3] 的方法以及 Farid [6] 的方法, 偵測的可靠度較差而且無法估計嵌入訊息的大小; Westfeld 與 Pfizman [14] 的方法能夠估計嵌入訊息的大小, 但卻只適用於嵌入像素位置是固定而且連續的情形。然而, Fridrich 等人 [7, 8] 的方法, 稱為 RS (Regular and Singular) 分析法, 則不論嵌入像素位置是連續的或是隨機選取的均可適用, 偵測的可靠度及靈敏度也較高, 還能夠相當可靠的估計出嵌入訊息的大小, 是目前為止針對 LSB 嵌入法的最有效的攻擊法。因此, 新近提出的 LSB 嵌入法都須要求能夠抵擋 Fridrich 等人所提出的 RS 統計偵測攻擊法。

2003 年, Wu 與 Tsai [16] 提出一個根據相鄰的兩個像素所組成的區塊之平滑級別來嵌入不同數量的訊息位元的嵌入作法 (在較平滑的區塊嵌入較少量的訊息; 反之則嵌入較多的訊息), 同時根據實驗結果指出他們的方法不但具有良好的不可察覺性而且可以抵擋 RS 統計偵測的攻擊。所以, Wu 與 Tsai [16] 的嵌入法可算是一個相當安全的影像偽裝技術。然而, 因為對一般的自然影像而言, 相鄰的兩個像素之值通常差異不大, 因此據以嵌入之訊息量也就難以令人滿意。有鑒於一個真正實用有效的影像偽裝術, 除了要具備難以察覺與難以偵測的安全性外, 尚須要求具有足夠抵擋自然雜訊干擾的強韌性、盡可能高的嵌入容量、以及盡可能低的計算複雜度; 本論文參考 Wu 與 Tsai [16] 的作法提出一個改良的嵌入技術。首先, 將掩護影像分割為若干個相同大小的區塊並計算每一區塊的平均值, 然後根據區塊中的每一像素與平均值之差異級別來決定嵌入每一像素的訊息量, 使得平滑度較小的區塊得以嵌入較多的訊息; 反之, 平滑度較大的區塊則嵌入較少的訊息。在嵌入處理時, 可以事先判斷每一像素是否適合用來嵌入秘密訊息; 在萃

取處理時，則不必參考原始掩護影像而且可以事先判斷每一像素是否含有嵌藏訊息。接著，我們進行一系列的實驗。實驗結果證明，本論文的方法與 Wu 與 Tsai[16]的方法一樣的，也具有相當良好的不可察覺性並且可以抵擋 Fridrich 等人[7, 8]的 RS 統計偵測攻擊。最後，實驗的結果進一步顯示，在安全的前提下，本論文的作法可以比 Wu 與 Tsai[16]的方法嵌入更多的訊息位元。因此，我們的方法更適於偽裝術的實際應用。

2. RS 統計偵測法之回顧

本節將以一個灰階無失真壓縮影像格式圖檔(即 BMP 格式)來說明 Fridrich 等人[7, 8]所提出的 RS 偵測技術的原理及作法。

Fridrich 等人[7, 8]先將影像中每 n 個相鄰的像素當作一個像素群組 (group) $G = (x_1, x_2, \dots, x_n)$ ，其中每一像素值 $x_i \in \{0, 1, 2, \dots, 255\}$ 。接著，採用兩種函數，鑑別函數 (discrimination function) f 與翻轉函數 (flipping function) F ，來將所有群組歸類為規則 (regular) 群組、奇異 (singular) 群組以及無用 (unusable) 群組。鑑別函數 f 的目的是為了獲得像素群組 G 的平滑性 (smoothness)，若像素群組 G 中所含的雜訊越多(例如嵌入該群組的訊息位元愈多的時候)，則像素群組的鑑別函數值便會增加。例如，若以群組中各像素值的差異 (variation) 作為群組平滑性的量度，則鑑別函數 f 可以定義如下：

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|, 0 \leq x_i \leq 255$$

。

翻轉函數 F 的目的在於模擬將一像素的 LSB 位元作翻轉(即 $0 \leftrightarrow 1$)的操作。它通常是像素值 $\{0, 1, 2, \dots, 255\}$ 的一個 2-循環排列；所以具有 $F(F(x)) = x$ ， $x_i \in \{0, 1, 2, \dots, 255\}$ ，或 $F^2 = \text{Identity}$ 之特性。RS 偵測技術使用了下列三種翻轉函數：

$$(1) \text{ 翻 轉 函 數 } F_1 : \\ 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 252 \leftrightarrow 253, 254 \leftrightarrow 255$$

$$(2) \text{ 平 移 翻 轉 函 數 } F_{-1} : \\ -1 \leftrightarrow 0, 1 \leftrightarrow 2, \dots, 253 \leftrightarrow 254, 255 \leftrightarrow 256 \\ \text{, 或是表示為 } F_{-1}(x) = F_1(x+1) - 1$$

$$(3) \text{ 不 翻 轉 函 數 } :$$

$$F_0(x) = x, \forall x \in \{0, 1, \dots, 255\}。$$

根據鑑別函數 f 與翻轉函數 F ，RS 偵測法將所有的像素群組區分為以下三種類型：

$$(1). \text{ 規 則 群 組 } R \text{ (Regular Group)} :$$

$$G \in R \Leftrightarrow f(F(G)) > f(G)$$

$$(2). \text{ 奇 異 群 組 } S \text{ (Singular Group)} :$$

$$G \in S \Leftrightarrow f(F(G)) < f(G)$$

$$(3). \text{ 無 用 群 組 } U \text{ (Unusable Group)} :$$

$$G \in U \Leftrightarrow f(F(G)) = f(G)。$$

其中 $F(G) = (F(x_1), F(x_2), \dots, F(x_n))$ 表示利用翻轉函數 F 對群組 G 的每一像素進行翻轉處理， $f(F(G))$ 表示翻轉處理後的鑑別函數 f 值。我們也可以對群組 G 的不同像素進行不同的翻轉操作。此時，可令 $M = [M(1), M(2), \dots, M(n)]$ 為一翻轉遮罩 (flipping mask)， $M(i) \in \{-1, 0, 1\}$ ，則 $F_M(G) = (F_{M(1)}(x_1), F_{M(2)}(x_2), \dots, F_{M(n)}(x_n))$ 並且以 R_M 與 S_M 分別代表翻轉處理後，規則群組與奇異群組佔全部群組的比率。例如，假設 $G = (x_1, x_2, x_3, x_4)$ 且 $M = [0, 1, 1, 0]$ ，則 $F_M(G) = (F_0(x_1), F_1(x_2), F_1(x_3), F_0(x_4))$ 使用翻轉函數 F_0 來處理 x_1 與 x_4 ，使用翻轉函數 F_1 來處理 x_2 與 x_3 。

Fridrich 等人[7, 8]指出，在一般未經藏密的自然影像(例如掩護影像)中， R_M 與 R_{-M} 的期望值近似相等， S_M 與 S_{-M} 的期望值亦然：

$$R_M \cong R_{-M} \text{ 與 } S_M \cong S_{-M} ; \quad (2-1)$$

但是，隨機的翻轉各像素的 LSB 位元後(例如在各像素的 LSB 位元嵌入雜訊) 便會破壞此一統計特性。RS 偵測法就是據此來偵測 LSB 嵌入訊息的存在性。

RS 偵測法一：偵測嵌入訊息的存在

輸入：一個特定的嵌入作法 E ，與一個掩護影像 I 。

輸出：0 或 1(0 表無法偵測到嵌入法 E 所產生的偽裝影像中有秘密訊息之存在；1 表反是)。

步驟：1. 利用 E 嵌入法，以 5% 的增量依次嵌入不同百分長度的訊息到受測影像 I 中。

//註：嵌入訊息的百分長度 = (嵌入的位元數 / 掩護影像的像素數) × 100% //

2. 使用翻轉遮罩 $M=[0, 1, 1, 0]$ 與 $-M=[0, -1, -1, 0]$ 對每次嵌入訊息後的影像計算 R_M 、 R_{-M} 、 S_M 、與 S_{-M} 之值，同時檢驗(2-1)式的統計假設是否成立。
3. 如果每次嵌入訊息後，步驟 2 的檢驗都成立，則輸出 0；否則輸出 1。

例如，令 E 為傳統的空間域 LSB 嵌入法(即將掩護影像在空間域的某些連續或隨機選取的像素的 LSB 位元由祕密訊息位元來取代)且 I 為一 512×512 的灰階 Lena 影像。若依據上述步驟進行嵌入與偵測的實驗並繪製 R_M 、 R_{-M} 、 S_M 、與 S_{-M} 之圖形(稱為 RS 曲線圖)，結果如圖 2.1 所示：一旦嵌入任何數量的雜訊，(2-1)式的統計假設 $R_M \cong R_{-M}$ 與 $S_M \cong S_{-M}$ 便不成立，RS 法可以立即偵測出嵌入訊息的存在性。

RS 偵測法可以進一步的估計出 LSB 嵌入訊息的百分長度，其所根據的原理與作法如下。令 $M=[M(1), M(2), \dots, M(n)]$ 為一翻轉遮罩且 $-M=[-M(1), -M(2), \dots, -M(n)]$ 。令 $R_M(x)$ 、 $R_{-M}(x)$ 、 $S_M(x)$ 、與 $S_{-M}(x)$ 分別表示使用翻轉遮罩 M 或 $-M$ 在一原始掩護影像中模擬嵌入訊息使得 LSB 位元被翻轉之像素百分率為 x 時，規則群組與奇異群組佔全部群

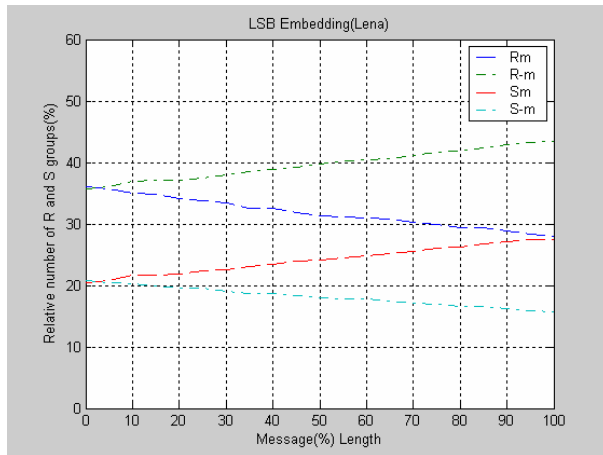


圖 2.1 傳統的空間域 LSB 嵌入法的 RS 偵測曲線圖(x-軸表嵌入訊息的百分長度，y-軸表規則群組與奇異群組的百分數)

組的比率。Fridrich 等人[7, 8]經由實驗證明， $R_M(x)$ 、 $R_{-M}(x)$ 、 $S_M(x)$ 、與 $S_{-M}(x)$ 間具有如下的統計特性：

$$R_M(0) = R_{-M}(0) \text{ 與 } S_M(0) = S_{-M}(0) \quad (2-2)$$

以及

$$R_M(1/2) = S_M(1/2) \quad (2-3)$$

在[7, 8]中，Fridrich 等人進一步由實驗推導出 $R_M(x)$ 、 $R_{-M}(x)$ 、 $S_M(x)$ 、與 $S_{-M}(x)$ 之圖形(即 RS 曲線圖)如下，其中 $R_{-M}(x)$ 與 $S_{-M}(x)$ 的圖形近似於直線而 $R_M(x)$ 與 $S_M(x)$ 圖形則近似於拋物線。

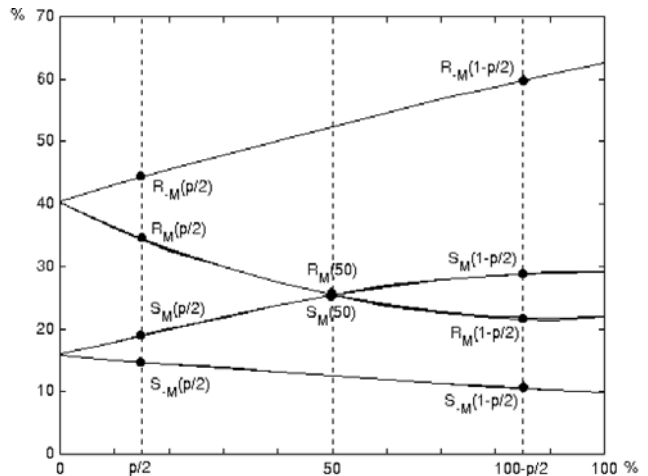


圖 2.2 RS 偵測曲線圖(x-軸表 LSB 位元被翻轉的像素百分率，y-軸表規則群組與奇異群組的百分數)

注意，因為每嵌入一個訊息位元到一個像素的 LSB 位元去的時候，該 LSB 位元會與不會被翻轉的機會各為 $1/2$ 。所以當實際嵌入的訊息百分長度(即嵌入的訊息位元數相對於影像的總像素數之百分比率)為 p 時，被翻轉之像素百分率約為 $p/2$ 。另外，因為 $R_M(0) = R_{-M}(0)$ 與 $S_M(0) = S_{-M}(0)$ ，所以 R_M 曲線與 R_{-M} 曲線之交點的 x -座標以及 S_M 曲線與 S_{-M} 曲線之交點的 x -座標均為 0；因為 $R_M(1/2) = S_M(1/2)$ ，所以 R_M 曲線與 S_M 曲線之交點橫座標為 $1/2 (=50\%)$ 。

根據式子(2-2)、(2-3)以及圖 2.2 之 RS 曲線圖，RS 偵測法估計 LSB 嵌入訊息的百分長度的作法如下[7, 8]。

RS 偵測法二：估計嵌入訊息的百分長度

輸入：以 LSB 法嵌入百分長度為 p 的訊息位元的偽裝影像 I_p (p 值未知)。

輸出： p 之估計值。

步驟：1. 直接由 I_p 計算規則群組與奇異群組的百分數，得出 $R_M(p/2)$ 、 $S_M(p/2)$ 、 $R_{-M}(p/2)$ 與 $S_{-M}(p/2)$ 之值。

//註：因為嵌入 I_p 的訊息百分長度為 p ，所以 I_p 中被翻轉之像素百分數為 $p/2$ 。//

2. 翻轉 I_p 的所有像素的 LSB 位元，然後計算規則群組與奇異群組的百分數，得出 $R_M(1-p/2)$ 、 $S_M(1-p/2)$ 、 $R_{-M}(1-p/2)$ 與 $S_{-M}(1-p/2)$ 之值。

//註：因為相對於原始的掩護影像， I_p 中已被翻轉的像素百分數為 $p/2$ ；所以翻轉 I_p 的所有像素後，因為對同一像素作連續兩次相同的翻轉時會產生抵消作用，所以實際被翻轉的像素百分數為 $1-p/2$ 。//

3. 計算下列諸數值：

$$d_0 = R_M(p/2) - S_M(p/2), \quad d_1 = R_M(1-p/2) - S_M(1-p/2)$$

$$d_{-0} = R_{-M}(p/2) - S_{-M}(p/2), \quad d_{-1} = R_{-M}(1-p/2) - S_{-M}(1-p/2)$$

。

4. 計算下列二次方程式的兩個根 z_1 與 z_2 ，並令其中絕對值較小者為 z ：

$$2(d_1 + d_0)z^2 + (d_{-0} - d_{-1} - d_1 - 3d_0)z + d_0 - d_{-0} = 0 \quad (2-4)$$

5. 計算並輸出 p 之估計值如下：

$$p = z / (z - 1/2) \quad (2-5)$$

//註：Fridrich 等人[7, 8]證明若令 $z = (x - p/2) / (1 - p)$ ，則 R_M 曲線與 R_{-M} 曲線之交點的 z -座標以及 S_M 曲線與 S_{-M} 曲線之交點的 z -座標會滿足(2-4)式。因此由(2-4)式求出交點的 z -座標後，再由 $z = (x - p/2) / (1 - p)$ 以及 $x=0$ 便可得出 $p = z / (z - 1/2)$ 。//

在參考文獻[7, 8]裡，Fridrich 等人經由實驗指出 RS 偵測法二可以相當可靠的估計出 LSB 法的嵌入訊息百分長度。在本論文裡，我們依據上述的作法進行實驗，結果與 Fridrich 等人的結論一致(參見第五節，表 5.4(a)與(b))。

3. Wu 與 Tsai 的偽裝技術[16]之回顧

2003 年，Wu 與 Tsai[16]提出一個灰階影

像的偽裝技術。他們的作法先將像素值域[0, 255]分段(或分級)如下：

$$[0, 255] = \bigcup_{i=1}^n R_i,$$

其中 $R_i = [\ell_i, u_i)$ ， $\ell_1 = 0$ ， $u_n = 256$ ， $u_i = \ell_{i+1}$ ，

而且 $|R_i| = (u_i - \ell_i) = 2^{r_i}$ ， $1 \leq i \leq n$ 。

接著，將掩護影像分割為由兩個像素組成的若干個區塊，再根據每一區塊的兩個像素值的差異級別來嵌入不同長度的訊息位元。其在掩護影像中嵌入訊息與從偽裝影像中萃取出嵌入訊息的作法可以簡略回顧如下。

訊息的嵌入演算法：

步驟：

- 1.(1) 將掩護影像分割為由兩個像素組成的若干個區塊。
- (2) 以一亂數種子與一擬亂數列產生器決定各個區塊的處理順序。
2. 依序對第 i 個區塊(假設其中兩個像素的灰階值為 g_i 和 g_{i+1})執行如下的訊息嵌入處理。

(1) (i) 計算 $d = g_{i+1} - g_i$

(ii) 決定 $|d|$ 所屬之區段 $R_k = [\ell_k, u_k)$ 。

(2) 令 $f(g_i, g_{i+1}, x)$

$$= \begin{cases} (g_i - \lfloor x/2 \rfloor, g_{i+1} + \lfloor x/2 \rfloor), & \text{當 } d \text{ 為奇數時;} \\ (g_i - \lfloor x/2 \rfloor, g_{i+1} + \lceil x/2 \rceil), & \text{當 } d \text{ 為偶數時。} \end{cases}$$

若 $d \geq 0$ 時 $f(g_i, g_{i+1}, u_k - d) \notin [0, 255]$

或 $d < 0$ 時 $f(g_i, g_{i+1}, -u_k - d) \notin [0, 255]$

則不使用此區塊來嵌入任何訊息，回到步驟 2 的開頭繼續處理下一個區塊。

//註：因為此時若仍嵌入訊息，則步驟(4)之 g'_i 和 g'_{i+1} 的值極有可能 $\notin [0, 255]$ //

否則執行步驟(iv)與(v)。

$$(3) \text{ 計算 } d' = \begin{cases} \ell_k + b, & d \geq 0; \\ -(\ell_k + b), & d < 0, \end{cases}$$

其中 b 為依序自秘密訊息中取出的 r_k ， $2^{r_k} = |R_k| = u_k - \ell_k$ ，個位元串之值。

(4) 修改 g_i 和 g_{i+1} 之值如下：

$$(g'_i, g'_{i+1}) = f(g_i, g_{i+1}, m)$$

$$= \begin{cases} (g_i - \lfloor m/2 \rfloor, g_{i+1} + \lfloor m/2 \rfloor), & \text{當 } d \text{ 為奇數時;} \\ (g_i - \lfloor m/2 \rfloor, g_{i+1} + \lceil m/2 \rceil), & \text{當 } d \text{ 為偶數時。} \end{cases}$$

其中, $m = d' - d$ 。

訊息的萃取演算法：

步驟：

- (1) 將掩護影像分割為由兩個像素組成的若干個區塊。
- (2) 使用與嵌入處理相同的亂數種子與擬亂數列產生器決定各個區塊的處理順序。
2. 依序對第 i 個區塊(假設其中兩個像素的灰階值為 g'_i 和 g'_{i+1})執行如下的訊息萃取處理。

(1) (i) 計算 $d' = g'_{i+1} - g'_i$

(ii) 決定 $|d'|$ 所屬之區段 $R_k = [\ell_k, u_k]$ 。

(2) 令 $f(g'_i, g'_{i+1}, x)$

$$= \begin{cases} (g'_i - \lfloor x/2 \rfloor, g'_{i+1} + \lfloor x/2 \rfloor), & \text{當 } d' \text{ 為奇數時;} \\ (g'_i - \lfloor x/2 \rfloor, g'_{i+1} + \lceil x/2 \rceil), & \text{當 } d' \text{ 為偶數時。} \end{cases}$$

若 $d' \geq 0$ 時 $f(g'_i, g'_{i+1}, u_k - d') \notin [0, 255]$

或 $d' < 0$ 時 $f(g'_i, g'_{i+1}, -u_k - d') \notin [0, 255]$

則回到步驟 2 的開頭繼續處理下一個區塊。//註：因為此區塊並未被使用來嵌入訊息//

否則執行步驟(3)。

(3) 取出嵌藏在第 i 個區塊訊息如下：

$$b = \begin{cases} d' - \ell_k, & d' \geq 0; \\ -(d' + \ell_k), & d' < 0. \end{cases}$$

在[16]中, Wu 與 Tsai 經由實驗證明他們的作法具有相當良好的不可察覺性並且可以抵擋 Fridrich 等人[7, 8]的 RS 統計偵測攻擊。然而, 因為對一般的自然影像而言, 相鄰的兩個像素之值通常差異不大, 因此據以嵌入之訊息量也就難以令人滿意。由於一個真正實用有效的影像偽裝術, 除了要具備難以察覺與難以偵測的安全性外, 尚須要求具有盡可能高的嵌入容量; 因此, 我們在下一節將參考 Wu 與 Tsai 的作法提出一個改良的嵌入技術。

4. 我們的作法

我們的演算法以灰階影像為掩護影像, 以位元串表示欲嵌入的秘密訊息, 並採用 Wu 與

Tsai[16]的作法將像素值域[0, 255]分段(或分

級)如下：

$$[0, 255] = \bigcup_{i=1}^n R_i,$$

其中 $R_i = [\ell_i, u_i]$, $\ell_1 = 0$, $u_n = 255$, $u_i = \ell_{i+1}$,

而且 $|R_i| = (u_i - \ell_i) = 2^{r_i}$, $1 \leq i \leq n$ 。

訊息的嵌入演算法：

步驟：

- (1) 將掩護影像分割成若干個大小同為 $m \times n$ (m 與 n 均為奇數)的區塊。
- (2) 以一亂數種子與一擬亂數列產生器決定各個區塊的處理順序。

2. 依序對每一區塊執行如下的訊息嵌入處理, 其中 t , $1 \leq t \leq m \times n$, 為區塊內之像素依照由上而下再由左而右之順序的編號, 且令 p_t 表示第 t 號像素的灰階值。

(1) 計算區塊內所有像素的平均值 μ 如下：

$$\mu = \left\lfloor \frac{\sum_{t=1}^{m \times n} p_t}{m \times n} \right\rfloor.$$

(2) 對每一 $t \neq \left\lfloor \frac{m \times n}{2} \right\rfloor$, 修改第 t 號像素

值 p_t 為 p'_t 如下：

(i) 計算 $d_t = \mu - p_t$ 。

(ii) 決定 $|d_t|$ 所屬之區段

$$R_k = [\ell_k, u_k]。$$

(iii) 令 $f(x, y) = x - y$ 。

若 $d_t \geq 0$ 時

$$f(p_t, u_k - d_t) \notin [0, 255]$$

或 $d_t < 0$ 時

$$f(p_t, -u_k - d_t) \notin [0, 255]$$

則令 $p'_t = p_t$

//註：此表不嵌入任何訊息位元到第 t 號像素。//

否則執行步驟(iv)與(v)。

(iv) 計算 $d'_t = \begin{cases} \ell_k + b, & d_t \geq 0; \\ -(\ell_k + b), & d_t < 0, \end{cases}$

其中 b 為依序自秘密訊息中取出

的 r_k , $2^{r_k} = |R_k| = u_k - \ell_k$, 個位元串之值。

(v) 修改 p_t 之值如下：

$$p'_t = p_t - m_t,$$

其中 $m_t = d'_t - d_t$ 。

(3) 修改第 $\left\lceil \frac{m \times n}{2} \right\rceil$ 號像素值 $p_{\left\lceil \frac{m \times n}{2} \right\rceil}$ 為 $p'_{\left\lceil \frac{m \times n}{2} \right\rceil}$ 如下：

$$p'_{\left\lceil \frac{m \times n}{2} \right\rceil} = p_{\left\lceil \frac{m \times n}{2} \right\rceil} + \sum_{t \neq \left\lceil \frac{m \times n}{2} \right\rceil} m_t。$$

//註：如此可以保證偽裝區塊的像素平均值 $\mu' =$ 掩護區塊的像素平均值 μ ；而且保證 $d'_t = \mu' - p'_t$ 。//

根據以上的嵌入作法，掩護區塊與對應的偽裝區塊之間有如下的重要關係，據此我們可以很容易自偽裝影像中取出所嵌入的訊息。

1. 偽裝區塊的像素平均值 $\mu' =$ 掩護區塊的像素平均值 μ 。

$$\begin{aligned} (\because \sum_{t=1}^{m \times n} p'_t &= \sum_{t \neq \left\lceil \frac{m \times n}{2} \right\rceil} p'_t + p'_{\left\lceil \frac{m \times n}{2} \right\rceil}) \\ &= \sum_{t \neq \left\lceil \frac{m \times n}{2} \right\rceil} (p_t - m_t) + (p_{\left\lceil \frac{m \times n}{2} \right\rceil} + \sum_{t \neq \left\lceil \frac{m \times n}{2} \right\rceil} m_t) \\ &= \sum_{t=1}^{m \times n} p_t。 \end{aligned}$$

2. (1) 因為

(i) 當我們嵌入訊息位元到第 t 號像素時， $|d'_t|$ 愈大 $\Rightarrow |u - p'_t|$ 愈大 $\Rightarrow p'_t$ 愈有可能落在 $[0, 255]$ 之外，

(ii) 當 $d'_t \geq 0$ 時其最大可能值為 u_k ，當 $d'_t < 0$ 時其最小可能值為 $-u_k$ ，

(iii) $d_t > 0 \Leftrightarrow d'_t > 0$ ，

所以在嵌入步驟 2-(2)-(iii) 中，若 ($d_t \geq 0$ 且 $f(p_t, u_k - d_t) \notin [0, 255]$) 或 ($d_t < 0$ 且 $f(p_t, -u_k - d_t) \notin [0, 255]$) 成立，則捨棄第 t 號像素，不做任何嵌入動作。

(2) 因為 $f(x, y) = x - y$

$$\begin{aligned} \Rightarrow f(x, y_1 + y_2) &= x - (y_1 + y_2) = (x - y_1) - y_2 \\ &= f(x, y_1) - y_2 = f(f(x, y_1), y_2)， \end{aligned}$$

所以

$$f(p_t, u_k - d_t)$$

$$= f(p_t, (d'_t - d_t) + (u_k - d'_t))$$

$$= f(f(p_t, d'_t - d_t), u_k - d'_t)$$

$$= f(p'_t, u_k - d'_t)；$$

$$f(p_t, -u_k - d_t)$$

$$= f(p_t, (d'_t - d_t) + (-u_k - d'_t))$$

$$= f(f(p_t, d'_t - d_t), -u_k - d'_t)$$

$$= f(p'_t, -u_k - d'_t)。$$

(3) 由(2)可得當欲自偽裝影像中取出所嵌入的訊息時，若 ($d'_t \geq 0$ 時 $f(p'_t, u_k - d'_t) \notin [0, 255]$) 或 ($d'_t < 0$ 時 $f(p'_t, -u_k - d'_t) \notin [0, 255]$) 成立，則表示在第 t 號像素中並沒有任何嵌入訊息，不必執行任何萃取訊息的動作。

$$3. d'_t = \begin{cases} \ell_k + b, & d_t \geq 0; \\ -(\ell_k + b), & d_t < 0, \end{cases}$$

$$\Leftrightarrow b = \begin{cases} d'_t - \ell_k, & d'_t \geq 0; \\ -(d'_t + \ell_k), & d'_t < 0. \end{cases}$$

$$(\because d_t > 0 \Leftrightarrow d'_t > 0 \quad \text{而} \quad |d_t| \in R_k \Leftrightarrow |d'_t| \in R_k。)$$

$$4. d'_t = \mu' - p'_t。$$

$$\begin{aligned} (\because p'_t &= p_t - m_t = p_t - (d'_t - d_t) \\ &= (p_t + d_t) - d'_t = \mu - d'_t = \mu' - d'。) \end{aligned}$$

訊息的萃取演算法：

步驟：

- (1) 將偽裝影像分割成若干個大小同為 $m \times n$ (m 與 n 均為奇數) 的區塊。
- (2) 使用與嵌入處理相同的亂數種子與擬亂數列產生器決定各個區塊的處理順序。

2. 依序對每一區塊執行如下的訊息萃取處理，其中 $t, 1 \leq t \leq m \times n$ ，為區塊內之像素依照由上而下再由左而右之順序的編號，且令 p'_t 表示第 t 號像素的灰階值。

(1) 計算區塊內所有像素的平均值 μ' 如下：

$$\mu' = \left\lceil \frac{\sum_{t=1}^{m \times n} p'_t}{m \times n} \right\rceil。$$

(2) 對每一 $t \neq \left\lceil \frac{m \times n}{2} \right\rceil$ ，萃取出嵌藏在第

t 號像素之訊息如下：

(i) 計算 $d'_t = \mu' - p'_t$ 。

(ii) 決定 $|d'_t|$ 所屬之區段

$$R_k = [\ell_k, u_k)。$$

(iii) 令 $f(x, y) = x - y$ 。

若 $d'_t \geq 0$ 時

$$f(p'_t, u_k - d'_t) \notin [0, 255]$$

或 $d'_t < 0$ 時

$$f(p'_t, -u_k - d'_t) \notin [0, 255]$$

//表第 t 號像素並未嵌入任何訊息//
則回步驟(2)，繼續處理下一個像素。

否則執行步驟(iv)。

(iv) 取出嵌藏在第 t 號像素之訊息如下：

$$b = \begin{cases} d'_t - \ell_k, & d'_t \geq 0; \\ -(d'_t + \ell_k), & d'_t < 0. \end{cases}$$

為了進一步說明我們的演算法，讓我們以一個 3×3 的掩護區塊

154 159 161

156 157 160

以及如下表 $(158, 161, 163, 0, 2), R_2=[2, 4), R_3=[4, 8), R_4=[8, 12), R_5=[12, 16), R_6=[16, 24), R_7=[24, 32), R_8=[32, 48), R_9=[48, 64), R_{10}=[64, 96), R_{11}=[96, 128), R_{12}=[128, 192), R_{13}=[192, 256)$ ，為例來嵌入秘密訊息 $M=1100,0000,00$ 如下：

(1) 計算區塊內所有像素的平均值

$$\mu = \left\lfloor \frac{\sum_{t=1}^9 p_t}{9} \right\rfloor = \lfloor 158.777 \rfloor = 158。$$

(2) (i) 計算 $d_1 = \mu - p_1 = 158 - 154 = 4$ 。

(ii) $|d_1| = 4 \in R_3 = [4, 8)$ 。

(iii) 令 $f(x, y) = x - y$ 。因為 $d_1 \geq 0$ 而

$$f(p_1, 8 - d_1) = 154 - (8 - 4)$$

$$= 150 \in [0, 255]$$

，表第 1 號像素可以用來嵌入秘密訊息。

(iv) 因為 $|R_3| = 8 - 4 = 4 = 2^2$ ，故可嵌入 2

個訊息位元。自訊息 M 中取出前面

2 個位元 11，並計算

$$d'_1 = \ell_3 + b = 4 + 11_2 = 4 + 3 = 7。$$

(v) 修改 p_1 之值如下：

$$m_1 = d'_1 - d_1 = 7 - 4 = 3，$$

$$p'_1 = p_1 - m_1 = 154 - 3 = 151。$$

(3) 同理可以得知此區塊的每一像素(第 5 號除外)都可以用來嵌入秘密訊息，且經過嵌入處理後每一像素的新值可以計算如下：

$$d_2=2, d_3=0, d_4=-1, d_6=-3,$$

$$d_7=-3, d_8=-2, d_9=-5;$$

$$d'_2=2, d'_3=0, d'_4=0, d'_6=-2,$$

$$d'_7=-2, d'_8=-2, d'_9=-4;$$

$$m_2=0, m_3=0, m_4=1, m_6=1,$$

$$m_7=1, m_8=0, m_9=1;$$

$$p'_2=156, p'_3=158, p'_4=158,$$

$$p'_6=160, p'_7=160, p'_8=160,$$

$$p'_9=162。$$

(4) $\because \sum_{t \neq 5} m_t = 3 + 0 + 0 + 1 + 1 + 1 + 0 + 0 + 1 = 7$

$$\therefore p'_5 = p_5 + 7 = 164。$$

(5) 由(1)~(4)可得偽裝區塊如下：

151 158 160

156 164 160

158 160 162

接著，根據我們的萃取演算法可以從上面的偽裝區塊取出嵌入的訊息如下：

(1) 計算偽裝區塊內所有像素的平均值 μ' 如下：

$$\mu' = \left\lfloor \frac{\sum_{t=1}^9 p'_t}{9} \right\rfloor = \lfloor 158.777 \rfloor = 158。$$

(2) (i) 計算 $d'_1 = \mu' - p'_1 = 158 - 151 = 7$ 。

(ii) $\because |d'_1| = 7 \in R_3 = [4, 8)$ ，

(iii) 令 $f(x, y) = x - y$ 。因為 $d'_1 \geq 0$ 而

$$f(p'_1, 8 - d'_1) = 151 - (8 - 7) = 150 \in [0, 255]$$

，表第 1 號像素含有嵌入訊息，必須進行訊息萃取的動作。

(iv) 取出嵌藏在第 1 號像素之訊息如下：

$$b = d'_1 - \ell_3 = 7 - 4 = 3 = 11_2。$$

(3) 類似的，由 d'_t , $2 \leq t \leq 9$ 且 $t \neq 5$ ，可以取出其餘的嵌入訊息。

由上面的例子可以得知我們的演算法觀念簡單、計算容易、嵌入處理時可以事先判斷每一像素是否適合用來嵌入秘密訊息、萃取處理時不必參考原始掩護影像而且可以事先判斷每一像素是否含有嵌藏訊息。因為我們的作法採用較大的區塊並利用區塊內各個像素值與整個區塊像素平均值的差異程度來嵌入適量的訊息，顯然的，所允許的嵌入量應該會比 Wu 與 Tsai[16]的作法為大。至於偽裝影像的安全性（不可覺察性與不可偵測性），因為我們的作法是以區塊平均值為基準來修改掩護區塊的個別像素值，並使偽裝前後的區塊平均值維持不變（也即掩護區塊的平均值＝偽裝區塊的平均值），故應可維持偽裝影像的品質並對一般的統計偵測攻擊具有一定程度的抵抗能力。事實上，經由一系列的實驗和結果，可以證明我們的方法的確具有如上的良好特性。我們將在下一節詳細的說明我們的實驗作法與結果。

5. 實驗與結果

在本節裡，我們將經由一系列的實驗來證明我們的方法與 Wu 與 Tsai[16]的作法一樣的，具有相當良好的不可察覺性並且可以抵擋 Fridrich 等人的 RS 統計偵測攻擊[7, 8]。此外，在安全的前提下，具有比 Wu 與 Tsai[16]的作法更高的嵌入容量。我們實驗所使用的掩護影像、嵌入訊息、像素值域分段、以及使用的軟體設備如下：

掩護影像：包括 512×512 的灰階 Lena、Jet、Peppers, Baboon 以及 Barbara 等影像(見圖 5.1)。表 5.1 為各掩護影像利用 RS 法偵測所得的雜訊量(以訊息百分長度表示)。

圖 5.1(置於參考文獻之後)

表 5.1(置於參考文獻之後)

嵌入訊息：中英對照聖經—撒迦利亞書的內容。

像素區塊大小：1×3 與 3×3 兩種。

像素值域之分段： $R_1=[0, 2)$, $R_2=[2, 4)$, $R_3=[4, 8)$, $R_4=[8, 12)$, $R_5=[12, 16)$, $R_6=[16, 24)$, $R_7=[24, 32)$, $R_8=[32, 48)$, $R_9=[48, 64)$, $R_{10}=[64, 96)$, $R_{11}=[96, 128)$, $R_{12}=[128, 192)$, $R_{13}=[192, 256)$ 。

軟體：Matlab 6.5。

硬體：Acer TravelMate370，CPU：Intel Pentium 1.4G，RAM：768MB。

實驗 1：最大的嵌入容量 (Maximum embedding capacity)

內容：分別計算當掩護影像的每一像素區塊都恰好使用一次時，LSB 嵌入法、Wu 與 Tsai[16]的嵌入法、以及我們的嵌入法所能嵌入的最多訊息位元數。

結果：不論使用 1×3 的區塊或是 3×3 的區塊，我們的嵌入法都比 Wu 與 Tsai[16]的嵌入法具有較大的嵌入容量(見表 5.2)。

表 5.2(置於參考文獻之後)

實驗 2：不可察覺性 (Imperceptibility)

內容：分別利用 LSB 嵌入法、Wu 與 Tsai 的嵌入法[16]、以及我們的嵌入法以 5%的增量依次嵌入不同百分長度的訊息到各掩護影像，然後計算其 PSNR 值與 RMSE 值以比較不同作法的不可察覺性。

結果：我們的嵌入法之 PSNR 值雖然略小於 LSB 嵌入法以及 Wu 與 Tsai[16]的嵌入法，但是在最大嵌入容量下仍然具有 36 以上的 PSNR 值。因此，攻擊者很難以肉眼察覺出秘密訊息的存在(見表 5.3(a)與(b))。

表 5.3(a)與(b) (置於參考文獻之後)

實驗 3：不可偵測性 (Non-detectability)

內容：(1)利用我們的嵌入法，以 5%的增量依次嵌入不同百分長度的訊息到 Lena

掩護影像，並對每次嵌入後的偽裝影像以 RS 偵測法計算並檢驗(2-1)式的統計假設 $R_M \cong R_{-M}$ 與 $S_M \cong S_{-M}$ 是否成立。如果成立，則表示 RS 偵測法無法偵測出我們的偽裝影像中有秘密訊息的存在。

(2)分別利用 LSB 嵌入法、Wu 與 Tsai[16] 的嵌入法、以及我們的嵌入法以 5% 的增量依次嵌入不同百分長度的訊息到各掩護影像，並對每次嵌入後的偽裝影像以(2-5)式估計嵌入訊息的百分長度，來比較不同作法能夠抵擋 RS 偵測攻擊的程度。

結果：(1) RS 偵測法無法偵測出我們的偽裝影像中有秘密訊息的存在(見圖 5.2-(a)~(d))。

圖 5.2-(a)~(d) (置於參考文獻之後)

(2) RS 偵測法無法有效的估計我們的作法實際嵌入的訊息的百分長度，也無法由之判斷我們的偽裝影像中是否有秘密訊息的存在(見表 5.4(a)與(b))。

表 5.4(a)與(b) (置於參考文獻之後)

6. 結論

由於一個真正實用有效的影像偽裝術，除了要具備難以察覺與難以偵測的安全性外，尚須要求具有盡可能高的嵌入容量；本論文參考 Wu 與 Tsai[16]的作法提出一個改良的影像偽裝技術。首先，將掩護影像分割為若干個相同大小的區塊並計算每一區塊的平均值，然後根據區塊中的每一像素與平均值之差異級別來決定嵌入每一像素的訊息位元量，使得平滑度較小的區塊得以嵌入較多的訊息；反之，平滑度較大的區塊則嵌入較少的訊息。接著，我們進行一系列的實驗來比較傳統的 LSB 嵌入法、Wu 與 Tsai[16]的嵌入法、以及我們的作法的安全性與嵌入容量。實驗結果證明，我們的方法具有如下的優點：

1. 具有良好的不可察覺性。
2. 能夠有效的抵擋 Fridrich 等人[7, 8] 於 2001 年所提出 RS 統計偵測攻擊。

3. 在安全的前提下，具有比 Wu 與 Tsai[16] 法更高的訊息嵌入量。
4. 嵌入處理時可以事先判斷每一像素是否適合用來嵌入秘密訊息。
5. 萃取處理時不必參考原始掩護影像，而且可以事先判斷每一像素是否含有嵌藏訊息，是否須要執行萃取計算。
6. 觀念簡單並且計算容易。

參考文獻

- [1] M. M. Amin, M. Salleh, S. Ibrahim, M. R. Katmin, and M. Z. I. Shamsuddin, "Information Hiding using Steganography," IEEE National Conference on Telecommunication Technology, pp. 21-25, 2003.
- [2] R. J. Anderson, and F. A. Petitcolas, "On the Limits of Steganography," IEEE Journal of Selected Areas in Comm., Vol. 16, No. 4, pp. 474-481, 1998.
- [3] I. Avcibas, ND. Memon and B. Sankur, "Steganalysis based on Image Quality Metrics," Proc. Of the IEEE 4th Workshop on Multimedia Signal Processing. Cannes:IEEE, 2001, pp.517-522.
- [4] W. Bender, D. Gruhl, N. Morimoto, and A. Liu, "Techniques for Data Hiding," IBM System Journal, Vol. 35, No. 3&4, pp. 313-336, 1996.
- [5] 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.
- [6] H. Farid, "Detecting Hidden Messages

- Using Higher-Order Statistical Models,” Proc. Of the Ieee Int’l Conf. on Image Orocessing 02, Vol.2, New York:IEEE, 2002, pp.905-908.
- [7] J. Fridrich, M. Goljan, and R. Du, “Reliable Detection of LSB Steganography in Color and Grayscale Images,” Proc. of the ACM Workshop on Multimedia and Security, CA, pp. 27-30, 2001.
- [8] J. Fridrich, and M. Goljan, “Practical Steganography of Digital Images: State of the Art,” Proceedings of SPIE Photonics West, pp. 1-13, 2002.
- [9] J. Fridrich, M. Goljan, and R. Du, “Detecting LSB Steganography in Color and Gray-Scale Images,” Magazine of IEEE Multimedia Special Issue on Security, pp. 22-28, 2001.
- [10] C. T. Hsu and J. L. Wu, “Hidden digital watermarks in image,s” IEEE Transactions on Image Processing, Vol. 8, No. 1, pp. 58-68, 1999.
- [11] S. Katzenbeisser and F. A. Petitcolas, Information Hiding: Techniques for Steganography and Digital Watermarking, Norwood, MA: Artech House, 2000.
- [12] N. Nikolaidis and I. Pitas, “Robust image watermarking in the spatial domain,” Signal Process. 66 (3) , pp. 385-403 , 1998.
- [13] S. K. Pal, P. K. Saxena and S. K. Muttoo, “A Systematic Approach to Steganalysis of Images,” Proc. of the Pacific Rim Workshop on Digital Steganography, Kitakyushu Kyushu Institute of Technology, 2002, pp.2-188-200.
- [14] A. Westfeld and A. Pfitzmann, “Attacks on Steganographic Systems,”Proc. Third Internat. WorkshopInformation Hiding. In: Lecture Notes in Computer Science, Vol. 1768. Springer-Verlag, Berlin, pp.61-75.
- [15] D. C. Wu, and W. H. Tsai, “Spatial-domain image hiding using image differencing,” Vision, Image and Signal Processing, IEE Proceedings, Vol. 147, Issue: 1, pp. 29-37, 2000.
- [16] D. C. Wu, and W. H. Tsai, “A Steganographic Method for Images by Pixel-Value Differencing,” Pattern Recognition Letters, Vol. 24, pp. 1613-1626, 2003.



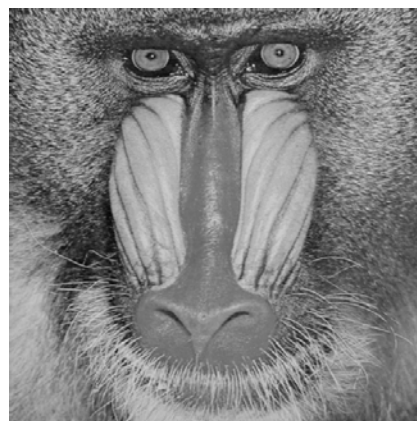
(a) Lena



(b) Jet



(c) Peppers



(d) Baboon



(e) Barbara

圖 5.1 掩護影像 512×512 灰階(a) Lena (b) Airplane (c) Peppers (d) Baboon (e)Barbara 影像

表 5.1 掩護影像利用 RS 法偵測所得的雜訊量(以訊息百分長度表示)

掩護影像	Lena	Jet	Pepper	Baboon	Barbara	Average
雜訊量	2.35%	2.26%	0.18%	0.32%	1.59%	1.34%

表 5.2 LSB 嵌入法、Wu 與 Tsai 的嵌入法、以及我們的嵌入法的最大嵌入容量

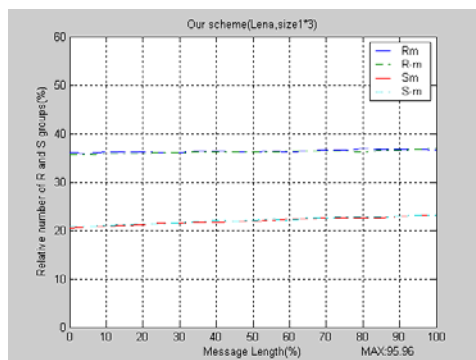
掩護影像	LSB 法	Wu 與 Tsai 法	我們的方法	
			Block size:1×3	Block size:3×3
Lena	262,144	185,910	251,548	337,408
Jet	262,144	187,128	233,322	333,132
Pepper	262,144	192,870	236,615	331,993
Baboon	262,144	308,620	337,669	500,674
Babara	262,144	241,281	315,793	440,196

表 5.3(a) Lena 影像

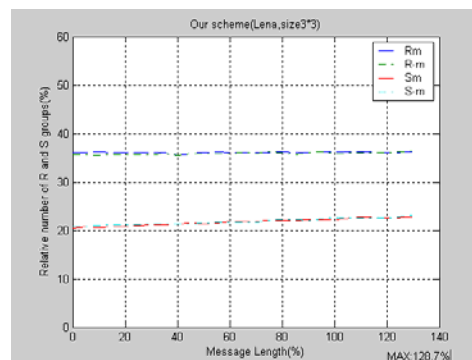
嵌入訊息 百分長度	LSB 法		Wu 與 Tsai 法		我們的方法 (Block size:1×3)		我們的方法 (Block size:3×3)	
	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
0%	-	-			-	-	-	-
5%	0.1588	64.1155	0.2213	61.2322	0.3974	56.1468	0.4048	55.9864
10%	0.2239	61.1299	0.3218	57.9804	0.562	53.1256	0.6049	52.4969
15%	0.2736	59.3877	0.3949	56.2019	0.6986	51.2459	0.7075	51.1364
20%	0.3164	58.1257	0.4572	54.9284	0.8033	50.0329	0.8473	49.5700
25%	0.3537	57.1573	0.5151	53.8935	0.8925	49.1187	0.9380	48.6870
30%	0.3864	56.3911	0.5693	53.0246	0.9761	48.3410	1.0001	48.1302
35%	0.4192	55.6816	0.6123	52.3916	1.0593	47.6304	1.0965	47.3308
40%	0.4466	55.1322	0.6493	51.8825	1.1413	46.9826	1.1711	46.7589
45%	0.4733	54.6282	0.6922	51.3266	1.2023	46.5306	1.2398	46.2638
50%	0.4995	54.1604	0.7342	50.8140	1.2574	46.1416	1.3384	45.5991
55%	0.5242	53.7414	0.7682	50.4210	1.3186	45.7289	1.3801	45.3328
60%	0.5470	53.3704	0.8033	50.0332	1.3863	45.2934	1.4853	44.6945
65%	0.5706	53.0043	0.8385	49.6605	1.4548	44.8747	1.5192	44.4984
70%	0.5904	52.7086	0.8661	49.3798	1.4981	44.6200	1.5631	44.2509
(74.9%)			0.9006	49.0404				
75%	0.6119	52.3969			1.5381	44.3910	1.6265	43.9055
80%	0.6324	52.1108			1.5968	44.0657	1.6662	43.6961
85%	0.6522	51.8466			1.6475	43.7943	1.7346	43.3467
90%	0.6698	51.6114			1.7003	43.5200	1.7890	43.0786
95%	0.6896	51.3593			1.7389	43.3252	1.8110	42.9724
95.96%					1.7588	43.2265		
100%	0.7061	51.1538					1.8724	42.6826
110%							1.9761	42.2145
120%							2.0507	41.8928
128.7%							2.1317	41.5563

表 5.3(b) Baboon 影像

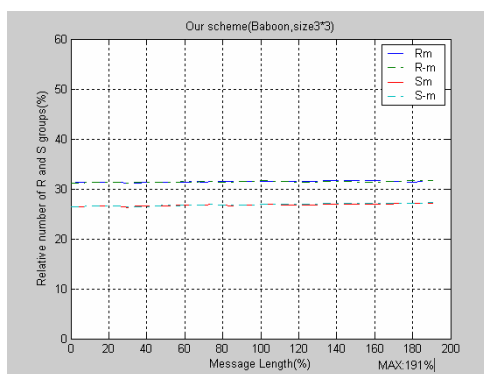
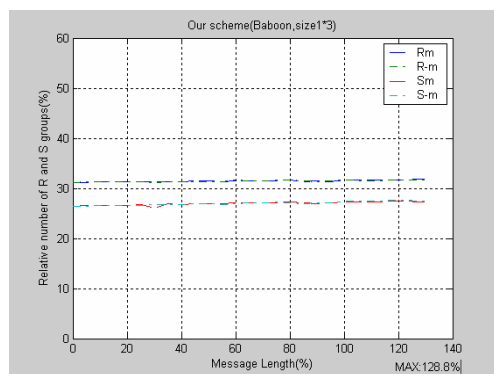
嵌入訊息 百分長度	LSB 法		Wu 與 Tsai 法		我們的方法 (Block size:1×3)		我們的方法 (Block size:3×3)	
	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR	RMSE	PSNR
0%			-	-				
5%	0.1579	64.1644	0.2900	58.8833	0.5090	53.9967	0.6311	52.1294
10%	0.2240	61.1266	0.4140	55.7913	0.7297	50.8675	0.9132	48.9195
15%	0.2737	59.3852	0.5119	53.9471	0.8631	49.4091	1.0817	47.4489
20%	0.3164	58.1274	0.6044	52.5044	1.0057	48.0814	1.2773	46.0052
25%	0.3520	57.1998	0.6686	51.6273	1.1295	47.0734	1.3996	45.2109
30%	0.3873	56.3708	0.7288	50.8790	1.2304	46.3301	1.5300	44.4370
35%	0.4191	55.6843	0.7736	50.3602	1.3424	45.5731	1.6804	43.6223
40%	0.4474	55.1161	0.8393	49.6522	1.4282	45.0351	1.8094	42.9799
45%	0.4752	54.5938	0.8915	49.1284	1.5163	44.5151	1.9184	42.4720
50%	0.5001	54.1499	0.9351	48.7139	1.6066	44.0125	1.9952	42.1310
55%	0.5247	53.7332	0.9793	48.3127	1.6797	43.6262	2.0759	41.7866
60%	0.5471	53.3693	1.0231	47.9324	1.7545	43.2480	2.1775	41.3716
65%	0.5706	53.0043	1.0653	47.5811	1.8363	42.8520	2.2782	40.9788
70%	0.5908	52.7022	1.1077	47.2421	1.8888	42.6070	2.3767	40.6114
75%	0.6137	52.3716	1.1380	47.0079	1.9554	42.3059	2.4690	40.2803
80%	0.6325	52.1097	1.1892	46.6258	2.0214	42.0176	2.5642	39.9518
85%	0.6521	51.8452	1.2248	46.3697	2.0798	41.7704	2.6030	39.8214
90%	0.6699	51.6102	1.2506	46.1886	2.1515	41.4758	2.6517	39.6603
95%	0.6899	51.3557	45.9061	45.9061	2.2108	41.2400	2.7572	39.3216
100%	0.7062	51.1524	1.3172	45.7381	2.2737	40.9963	2.8192	39.1283
102.9%			1.3372	45.6069				
103%			1.3372	45.6069				
110%					2.3748	40.6183		
120%					2.4812	40.2328	3.1159	38.2593
128.8%					2.5810	39.8952		
140%							3.3575	37.6104
160%							3.5815	37.0496
180%							3.8199	36.4898
191%							3.8679	36.3812



(a) 掩護影像：Lena，區塊大曉小：1×3



(b) 掩護影像：Lena，區塊大曉小：3×3



(c)掩護影像：Baboon，區塊大曉小：1×3

(d)掩護影像：Baboon，區塊大曉小：3×3

圖 5.2 我們的嵌入法的 RS 圖

表 5.4(a) RS 偵測法對不同嵌入法之嵌入訊息百分長度的估計值(掩護影像：Lena)

嵌入訊息 百分長度	RS 估計值			
	LSB 法	Wu 與 Tsai 法	我們的方法 (Block size:1×3)	我們的方法 (Block size:3×3)
0%	2.35%	2.35%	2.35%	2.35%
5%	4.09%	1.79%	0.63%	1.24%
10%	11.80%	2.82%	1.67%	3.04%
15%	13.65%	3.81%	1.59%	2.52%
20%	18.75%	1.16%	1.76%	2.50%
25%	23.11%	3.21%	0.15%	1.61%
30%	27.63%	2.68%	0.57%	1.88%
35%	36.22%	1.27%	1.65%	1.31%
40%	38.25%	2.19%	1.35%	1.43%
45%	45.81%	1.44%	0.85%	0.77%
50%	50.85%	1.24%	0.15%	1.11%
55%	54.30%	0.13%	1.67%	1.47%
60%	57.97%	1.23%	1.52%	0.28%
65%	61.55%	0.29%	0.36%	0.87%
70%	67.32%	1.18%	0.38%	0.97%
75%	71.45%	0.35%	0.99%	0.39%
80%	77.35%		3.61%	1.21%
85%	79.88%		1.91%	2.28%
90%	86.82%		0.26%	0.60%
95%	92.92%		0.42%	0.028%
100%	89.60%		1.65%	2.30%
MAX				

表 5.4(b) RS 偵測法對不同嵌入法之嵌入訊息百分長度的估計值(掩護影像：Baboon)

嵌入訊息 百分長度	RS 估計值			
	LSB 法	Wu 與 Tsai 法	我們的方法 (Block size:1×3)	我們的方法 (Block size:3×3)
0%	0.32%	0.32%	0.32%	0.32%
5%	2.75%	0.54%	1.49%	1.05%
10%	10.70%	0.91%	1.02%	1.16%
15%	14.34%	3.50%	1.23%	0.55%
20%	19.96%	3.41%	0.15%	1.00%
25%	26.85%	0.30%	0.31%	0.79%
30%	29.59%	2.12%	2.83%	3.55%
35%	36.33%	3.90%	3.50%	1.83%
40%	42.87%	3.11%	3.07%	0.49%
45%	47.67%	2.64%	0.14%	1.33%
50%	51.80%	2.79%	1.31%	1.01%
55%	57.77%	1.02%	0.65%	1.51%
60%	59.43%	4.97%	3.19%	0.68%
65%	69.72%	2.47%	2.06%	1.11%
70%	73.90%	1.73%	0.05%	0.39%
75%	73.87%	2.49%	0.58%	1.56%
80%	75.57%	3.95%	2.31%	2.43%
85%	86.83%	2.33%	1.90%	0.74%
90%	91.73%	0.66%	1.91%	0.26%
95%	91.06%	0.59%	2.62%	0.61%
100%	103.6%	1.18%	0.16%	0.68%
MAX				