階層式浮水印技術之失真控制機制

呂慈純
朝陽科技大學
資訊管理學系
tclu@cyut.edu.tw

許儷齡
朝陽科技大學
資訊管理學系
s9514641@cyut.edu.tw

摘要

由於網際網路及電腦科技的發達，為人們帶來了方便及快速。但也由於網路應用益趨普及的趨勢下帶來許多危機，例如，著作權侵犯、資料竊取和破壞等。因此，在資訊安全的領域下，資訊隱藏技術也越來越受到重視。為了解決這些問題，本篇研究擬提出一個有效的資訊隱藏技術，將驗證資訊、機密訊息等藏在數位媒體中以產生偽裝媒體。當收方收到此偽裝媒體時，可依之前嵌入進去的驗證資訊進行驗證，偵測該偽裝媒體是否因在網路上傳輸而遭竄改。此外，收方亦可在未經竄改的區塊中還原數位資訊的完整或者進行機密訊息的通訊。

關鍵詞：資訊隱藏、階層式浮水印、算術編碼法、無失真廣義最低有效位元嵌入法、驗證碼

1. 前言

近年來，因網際網路快速發展，為人類帶來方便快速的生活，然而，卻也衍生出許多潛在性的危機，例如，著作權侵犯、數位資料竄改等，資訊安全已經成為刻不容緩的議題。為了讓人們在網路上進行機密訊息傳輸時不被其他人察覺，進而防止資訊被破壞或竊取，資訊隱藏(Information Hiding)即為因應這些問題所產生的技術。

近年來已經有許多的資訊隱藏技術提出，例如最低有效位元取代法(Least Significant Bit Replacement)、量化導向隱藏方法(Quantization-based Embedding)、Tian 的可逆式差值擴張法(Reversible Data Embedding Using Difference Expansion)[7]等。上述的方法都是將秘密資訊藏於隱蔽媒體(Cover Image)之中，以產生另外一份偽裝媒體(Stego Image)。

數位產物的內容保護即稱為內容驗證，在許多方面的應用上都極為重要。正因目前已經有越來越多的媒體資訊在網路上傳遞皆採數位化的形式，內容驗證的議題更是不容忽視。由於數位媒體容易被複製及修改，使得智慧財產權的問題益加重要。學者提出了數位浮水印(Watermark)技術，用以驗證著作權[1]。本篇擬提出的方法是針對 Celik 等學者[11][10]所提之浮水印技術做改善，本所提方法將資訊藏匿於複雜度較高的區塊，為的是不讓非法第三者得知該圖片中藏有秘密資訊，如此一來，偽裝影像的品質會較 Celik 等學者所提出的方法還要高。

2. 相關文獻討論

本節中將探討與本研究相關的研究及技術。

2.1 浮水印技術

由於數位媒體在網路上傳輸時容易被複製及修改，因此，讓人不得不重視智慧財產權的問題，數位浮水印(Watermark)技術即應蘊而生[1]。

一般來說，畫家會在完成一幅作品後，在上面簽註自己的姓名，代表此幅作品是自己所

創作的，但是處於數位化的環境裡，這種簽名卻很容易被破壞或竄改，而數位浮水印就像是數位影像的簽名一樣，和傳統畫布上的簽名不同的是數位浮水印必需能夠抵抗各類型的破壞，進而達到保護智慧財產權[2]。

在已藏入數位浮水印的媒體中若被懷疑有竄改或盜用的嫌疑時，就可使用一公開的演算法來取出數位浮水印，以做為這個媒體的智慧財產權驗證。若從藏入數位浮水印的媒體外觀來看，數位浮水印技術可概分為兩大類，一類為可視(Visible)的浮水印技術，另一類為不可視的浮水印技術。顧名思義，可視的浮水印即是可明顯的被看見，類似紙鈔上的浮水印。然而可視性浮水印與紙鈔上的浮水印的目的卻不相同，紙鈔上的浮水印主要是用來防止被偽造及嚇阻盜印，而可視性浮水印的目的則是用來宣告此著作的所有權是受保護的。此種可視的浮水印不必經過任何運算就可辨視擁有者，但缺點則是會破壞該媒體的原有美觀性，而且加上去的浮水印很容易被不法第三者以訊號處理技術去除。現今的資訊技術發達，不法第三者可輕易的透過各種影像處理工具將數位媒體中的浮水印去除，甚至再加上別人的浮水印，原著者的簽章很容易遭到破壞。因此，在數位化環境中，可視的浮水印技術並不適用。第二類浮水印技術，即為不可視(Invisible)的技術，歸屬於資訊隱藏技術之一。不可視浮水印主要的特性除了無法直接從媒體外觀看出裡頭藏有浮水印外，還要能抵抗各種攻擊，例如旋轉、縮放等。

以設計藏入浮水印演算法的觀點來看，不可視的浮水印技術可區分為兩大類：第一類為空間域(Spatial Domain)，係藉由改變數位資料值來達到浮水印的藏入。此種方法的優點是運算速度快，但通常較難抵抗各類型訊號處理的破壞;另一類為頻率域(Frequency Domain)，係先將數位資料轉至頻率域，經轉換後，藉由轉換後所得到的係數值來藏入浮水印，再將之還原回原先的空間域，如此便完成浮水印的藏入，例如，傅立葉轉換(Fourier Transformation)、離散餘弦轉換(Discrete cosine Transformation)及小波轉換(Wavelet Transformation)等皆屬頻率域的方法。此種方法的優點為較易抵抗訊號處理的破壞，但相較之下，則比空間域的方法需要更多的運算[2]。

2.2 階層式浮水印技術 (Hierarchical Watermarking)

Celik 於 2006 年提出一個階層式浮水印[12]技術，用以維持偵測竄改範圍的準確性，在驗證過程中，採用階層式架構來做偵測[3]。階層式浮水印的架構如圖一所示。

我們用此架構來計算影像的驗證碼(Authentication Code, AC)。首先將影像切割為許多大小相同而且沒有重疊的區塊，每個區塊的最低階層，圖一中的 L_1 以 raster-scan 的方式讀出第 1 列所佔的位元數，我們稱此為訊息驗證碼(Message Authentication Code, MAC)[5]，之後選取某個區塊 X_1 切割為 4 個大小相同且不重疊的小區塊為 L_2。一樣使用用 raster-scan 方式讀入 X_2 中第 1 列所佔的位元數，此處讀入的位元數為第 1 階的 1/4，而第 3 階則是截取第 2 階中小區塊的其中一塊，同樣以 raster-scan 方式讀入第 1 列，此處讀入的位元數為第 1 階的 1/16。其階層式架構如圖一。

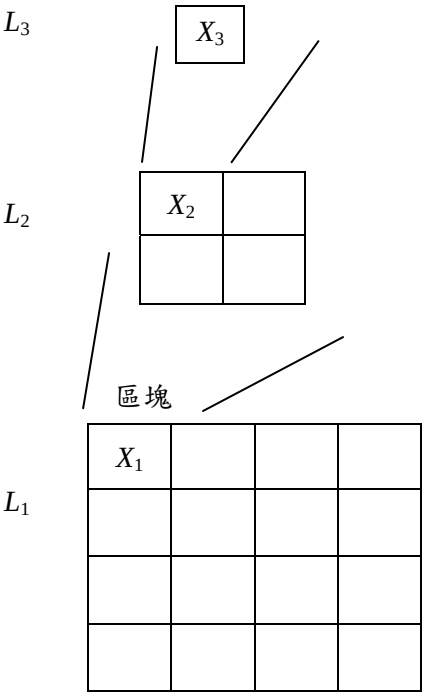

圖 一、階層式浮水印架構圖

因此驗證碼的公式為

$$AC=h(X_1)\|h(X_2)\|h(X_3)$$

其中的X_1指的是在第 1 階中任意選取的一個區塊，而h為一 MD5[13]的雜湊函數，函數的輸出為 128 位元的亂碼。我們將驗證碼以階層式浮水印的方式一層層的串在一起，得到最後的驗證碼用以偵測該影像中哪個區塊遭到竄改。

2006 年 Celik 等學者提出局部性驗證浮水印(Localized Lossless Authentication Watermark, L-LAW)，用以偵測圖形是否有被竄改，並且標示出被修改的位置。其中心概念就是以 MAC 為主要架構改良而成。然而，本研究發現，L-LAW 法有以下缺點：

(1) 無法控制資訊藏入的地方

例如圖二，深色部份是嵌入驗證碼、秘密資訊以及算術編碼的部份，而白色部份是未嵌入資訊的地方。如果白色部份太多則會造成資訊負載量過小，而且每個區塊的資訊負載量大小並不一定，那麼就無法控制資訊可藏入的地方。

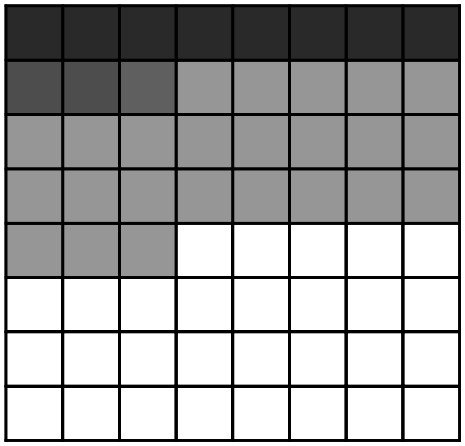

圖 二、已藏入資訊的區塊

(2) 失真狀況不易控制

倘若資訊正好嵌入在很平滑的的部份，造成失真狀況，則不法第三者較容易知道圖片中有藏入秘密資訊，而若藏在較複雜的部份，別人很難得知該張圖片有資訊的藏入。

為了改善上述的缺點，本篇論論文利用 Lossless G-LSB 方式提出一個能有效控制偽裝影像品質的資訊隱藏技術。下節將詳述 Lossless G-LSB 嵌入技術。

2.3 無失真廣義最低有效位元嵌入法(Lossless Generalized-LSB)

講述這個嵌入技術之前，必須先介紹一種無失真(Lossless Compression)的資料壓縮的技術，稱為算術編碼(Arithmetic Coding)[6]。算術編碼的輸出是一個實數且介於 0 到 1 之間，解碼端可以利用這個實數解碼回原來的訊息。我們必須先求出每個符號的出現機率，而後再繼續進行編碼的動作[4]。以一個例子說明，假設圖三(a)為一張原圖，經過量化後得到圖三(b)，其量化公式為$p'=\left\lfloor \frac{p}{q} \right\rfloor \times q$，$p$為原始像素，$q$為量化值，這個例子我們假設$q$=5。圖三(c)為原圖及量化的圖之差異，由圖中可以得知差異

值都介於 0 至 5 之間實數，因此我們依出現的比例做成 *R* 表。例如，0 出現 5 次，故其出現機率為 31.2%，如表一。

10	15	24	13
22	25	10	24
11	21	22	11
24	12	20	32

(a)原圖

10	15	20	10
20	25	10	20
10	20	20	10
20	10	20	30

(b)量化

(c)差異表(*R* 圖)

0	0	4	3
2	0	0	4
1	1	2	1
4	2	0	2

圖 三、差異量表實例

表 一、差異表比例分析(*R* 表)

R	比例	機率
0	5/16	31.2%
1	3/16	18.75%
2	4/16	25%
3	1/16	6.25%
4	3/16	18.75%

必須注意的是，每一個符號，即本例中的 *R* 值，所給定的範圍和其出現機率大小成正比，而每個符號都不包含範圍的最高值。建好 *R* 表後，就可以進行編碼的工作[4]。依 *R* 表的比例畫出區間圖，如圖四將 0 至 1 之區間做五等份切割。

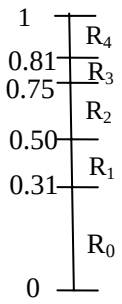

圖 四、比例區間圖

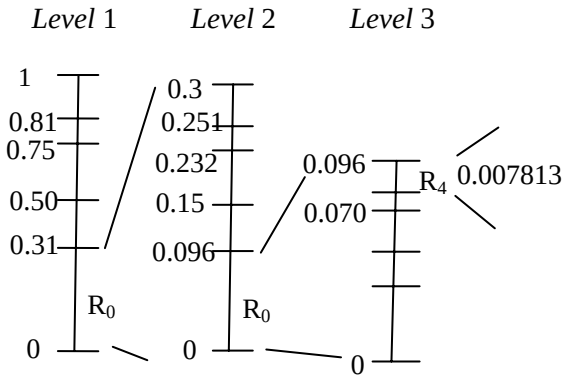

圖 五、 *R* 表的比例區間圖

接著，將圖三(c)以比例區間圖做壓縮，因為圖三(c)中第一個值為 0，對應到圖四中的 R_0 區間，接著，再將 R_0 做五等份切割，其上界為 0.31，下界為 0。圖三(c)中的第二個值亦為 0，故再從第二次切割的區間中找 R_0，依此類推，可自設欲做到第幾個值後停止。假設做到第 3 個值時停止，則到第三次切割的 R_4 中任意挑選一個值，若我們選定的值為$(0.007813)_{10}$，再將$(0.007813)_{10}$ 轉為二進值可得$(0.0000001)_2$ 為壓縮結果，小數點後的值稱為 Comp(*R*)，壓縮過程如圖五所示。之後將原本欲藏入的秘密資訊與 Comp(*R*)做串接的動作，即可得到欲嵌入的

資訊為「原本欲藏入的秘密資訊 ‖ Comp(*R*)」。接著再以無失真式 G-LSB 的方式做藏入。

無失真式 G-LSB(Lossless G-LSB)[9]嵌入法的計算方式和算術編碼的概念十分相像，不同的是 G-LSB 嵌入法在做區間切割時切割比例是固定的，在嵌入時將機密訊息投入逼近於它的區間中，再將量化後的值加上所屬區間號碼，即完成嵌入動作。承上例，假設原本欲藏入機密資訊為 1011，因此最後要嵌入的資訊為 10110000001。首先將這一串符號當做為二進制的小數點後部份，即可得$(0.10110000001)_2$，將它轉為十進制就變成$(0.68798828125)_{10}$。之後將這個十進制的值投入區間，找出最趨近於它的值，如下圖。

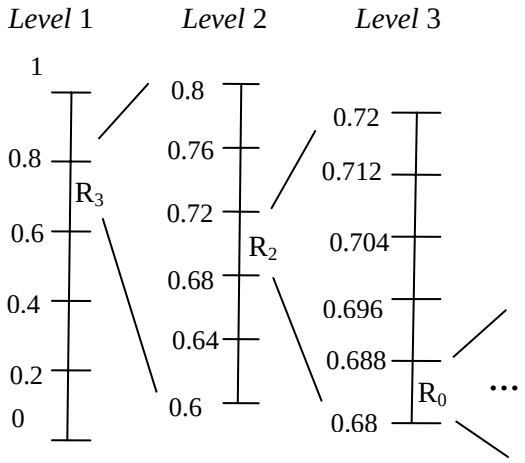

圖 六、G-LSB 區間圖

由圖六中能看出 *Level* 1 中最接近$(0.68798828125)_{10}$的區間為 0.6 至 0.8，位於 R_3 區間中，將其藏入圖三(b)中，因第一個像素量化後為 10，藏入時只需將 10 加上機密訊息對應的區間號碼即可，故偽裝像素為 10+3=13，依此類推。

Lossless G-LSB 嵌入法的取出過程，是將偽裝媒體先經過和發送端相同的量化之後，取出 *R* 圖，之後由 *R* 圖計算出 *R* 表，即可畫出比例區間圖，再將最後選定的十進制轉回二進制，就可得知嵌入的秘密資訊。

Lossless G-LSB 嵌入法即是將 G-LSB 嵌入法加上算術編碼的計算，因為 G-LSB 嵌入法並未記錄量化後圖與原圖區間的差值，如此便會造成影像無法正確還原，而 Lossless G-LSB 嵌入法則是因算術編碼記錄了差異值，因此在還原時可回復原始影像，並可取出所嵌入的秘密資訊。

3. 階層式浮水印技術之失真控制機制

為了解決 L-LAW 的缺點，本研究提出了一個改進的方法。所提方法先將影像切割為數個 8×8 的大區塊，而每個大區塊再切割為數個 2×2 的小區塊。針對這些小區塊計算變異數，若變異數愈大代表該區塊愈複雜，將資訊藏匿於複雜的區塊中，該影像在網路上傳輸時則較不容易被不法第三者懷疑影像中藏有秘密資訊。

計算完變異數之後，取出變異數最大的前四名區塊，並取出它們區塊內 16 個像素值的最低有效位元(LSBs)，然後將此 LSBs 嵌入至變異數排名為 5 至 16 的小區塊，而之前每個大區塊所選中的 16 個像素值將嵌入驗證碼，詳細嵌入步驟及取出步驟詳述於 3.1 節及 3.2 節。

3.1 嵌入步驟

資訊藏入的步驟如下：

步驟 1：將原圖切為數個 8×8 的大區塊，得到區塊 B。

步驟 2：每個區塊 B 切為數個 2×2 的小區塊，可得 b_0、b_1、b_2、b_3。

步驟 3：針對每個小區塊做量化，得到 $Q_L(b_i)$ 及 *R* 圖，量化公式為 $Q_L(b_i)=\left\lfloor \frac{b_i}{q} \right\rfloor \times q$，*R* 圖則為 b_i 與 $Q_L(b_i)$ 的差異值。

步驟 4：計算每個 $Q_L(b_i)$的變異程度。

步驟 5：將區塊依變化程度由大至小排序。

若變異程度愈大，代表此區塊的像素值之間變動得愈劇烈，因此在本研究稱此區塊為複雜區塊。用來判斷變化程度的計算式如下：

$$\sigma^2 = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2$$

其中 σ^2 為變異數 (Variance)，$x_k = x_1, x_2, ..., x_N$ 為區塊像素值，N 代表像素的個數，μ 則是平均值[14]。

步驟 6：取出計算後的變異數前四名所對應的小區塊的 LSBs，此為欲藏入的秘密資訊，取出之後將變異數前四名的原始 LSBs 改變為 0。

步驟 7：利用 Lossless G-LSB 方法藏匿資料。將步驟 6 中取出的 LSB 加上算術編碼，以 G-LSB 法將值嵌入至變異數第五至十六名的部份。

步驟 8：變異數的前 4 名所對應的小區塊用來藏匿驗證碼。

本研究針對一整個區塊 B 進行 MD5 的雜湊函數計算，可得到一串亂碼，再將此亂碼經假定為 16 位元的 HMAC[8]計算，得到 16 個二進制值，即為該區塊 B 的驗證碼。之後將此驗證碼依序嵌入至區塊 B 中變異數前四名的 LSBs 中，因為步驟 6 中將變異數前四名原始的 LSB 置為 0，因此可將驗證碼嵌入於其中。

以下我們舉個例子說明，圖七(a)為一張影像中的某一 8×8 區塊 B，且切成 2×2 的小區塊，圖七(b)為量化後的結果，圖七(c)為圖七(a)的像素值減圖七(b)量化後的像素值之差異量表。

54	6	125	90	84	105	28	0
57	24	47	90	125	68	13	12
10	52	39	89	92	0	89	97
19	79	48	29	36	57	82	101
21	14	54	66	80	123	38	9
59	26	35	50	44	88	28	81
32	91	68	115	118	53	117	18
56	7	110	94	1	40	53	9

(a)、區塊 *B*，並切成 2×2 的小區塊

50	5	125	90	80	105	25	0
55	20	45	90	125	65	10	10
10	50	35	85	90	0	85	95
15	75	45	25	35	55	80	100
20	10	50	65	80	120	35	5
55	25	35	50	40	85	25	80
30	90	65	115	115	50	115	15
55	5	110	90	0	40	50	5

(b)、量化

(c)、差異量表(*R* 圖)

4	1	0	0	4	0	3	0
2	4	2	0	0	3	3	2
0	2	4	4	2	0	4	2
4	4	3	4	1	2	2	1
1	4	4	1	0	3	3	4
4	1	0	0	4	3	3	1
2	1	3	0	3	3	2	3
1	2	0	4	1	0	3	4

圖 七、圖形實例

之後進行個別小區塊之變異數的計算，即可得圖八。圖九的是個別小區塊之變異數的排名，數字愈小代表其變異數愈大。

575	1075	706.25	106.25
941.6667	691.6667	1416.667	83.33333
375	150	1072.917	1006.25
1316.667	516.6667	2272.917	2472.917

圖八、小區塊的變異數

11	5	9	15
8	10	3	16
13	14	6	7
4	12	2	1

圖 九、計算後由大而小所排列的變異數名次

取出變異數前四名的 LSBs 得到 10110110 00010101，並將它們原來的 LSBs 變成 0，更改後的影像如圖十所示。

50	5	125	90	80	105	25	0
55	20	45	90	125	65	10	10
10	50	35	85	90	0	85	95
15	75	45	25	34	54	80	100
20	10	50	65	80	120	35	5
55	25	35	50	40	85	25	80
30	90	65	115	114	50	114	14
54	4	110	90	0	40	50	4

圖 十、修改後的區塊 *B*

依照每個符號出現的機率建立表二，並且依照表二的比例畫出區間圖，依序找出圖七(c)中每個差異值對應的區間為何，第一個差異值為 4，所以對應到 R_4，而第二個差異值為 1，對應到 R_1，依此類推。如圖十一。

表 二、差異表比例分析(*R* 表)

R	比例	機率
0	14/64	21.875%
1	10/64	15.625%
2	11/64	17.1875%
3	13/64	20.3125%
4	16/64	25%

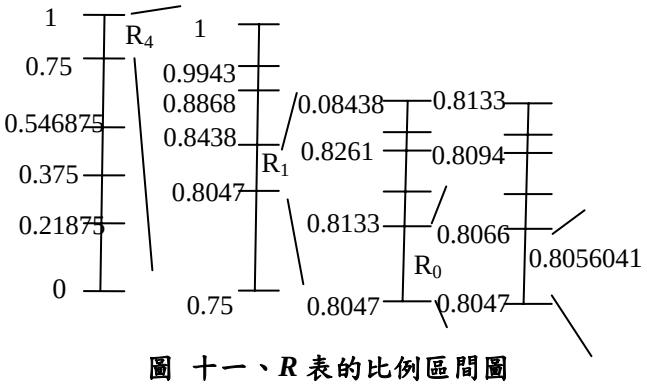

圖 十一、*R* 表的比例區間圖

假設我們以 4 個位元為計算單位，則最後從 R_0 區間 0.8047 至 0.8066 中選一數值。假設選擇的數值為$(0.8056041)_{10}$，將此十進制轉成二進制變成$(0.1100111001)_2$，此二進制值即為壓縮結果。小數點後的值為 Comp(*R*)，之後加上 LSBs 形成 1011011000010101 1100111001。將此值轉為十進制$(0.71127022802829)_{10}$，將這個十進制值投入區間，找出最趨近於它的值，如下圖。

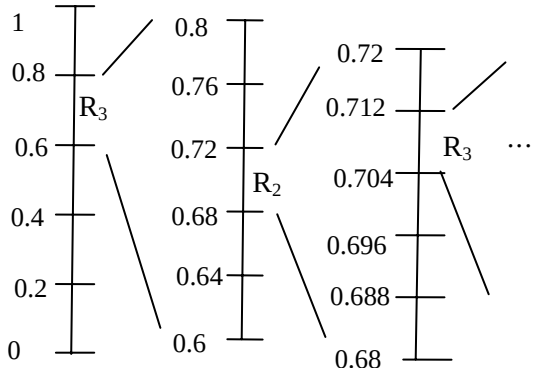

圖 十二 、G-LSB 區間圖

我們拿變異數為第五名，如圖七(b)中灰色部份，第一個值 125 加上機密訊息所對應的區間號碼，即 125+3=128，第二個值為 90+2=92，第三個值為 45+3=48，依此類推即可得偽裝像素。

而變異數前四名的 LSBs 所藏匿的是驗證碼，因此在其它變異數名次藏完資訊後，將區塊 B 進行 MD5 的雜湊函數及 HMAC 的計算，將所得的 16 個二進制值依變異數名次依序放置個別的 LSBs，假設最後得到 1001001000100010，則變異數的前四名改變後的值如圖十三，為偽裝區塊 B'。

50	5	128	92	80	105	25	0
55	20	48	94	125	65	10	10
10	50	35	85	90	0	85	95
15	75	454	25	35	54	80	100
20	10	50	65	80	120	35	5
55	25	35	50	40	85	25	80
30	90	65	115	114	50	115	14
55	4	110	90	1	40	50	5

圖 十三、偽裝區塊

暗色部份深淺即為變異數一至四名所對應的小區塊，所藏的是驗證碼，框起來的部份則是變異數第五名，藏匿了部份的秘密資訊，

3.2 取出步驟

在取出時，必須驗證每個區塊 B' 的驗證碼是否與發送端原先區塊 B 嵌入的驗證碼相同，如此一來，才能偵測哪些區塊已遭竄改，而哪些區塊與原先發送端所送出的區塊相同，倘若相同則再做該區塊的復原動作。

在經量化後，本研究最初在藏入資訊前的 $Q_L(b_i)$ 會等於取出時量化後的 $Q_L(b_i')$，因此可以計算出與發送端在嵌入資訊前相同的變異數名次。將變異數前四名的 LSBs 取出後做 MD5 與 HMAC 的計算，將計算後的值與變異數前四名的 LSBs 做比對，倘若比對不符，則表示此區塊 B' 已遭竄改。

資訊驗證及取出的步驟如下：

步驟 1：將原圖以 8×8 為單位切為數個大區塊，得到區塊 B'。

步驟 2：每個區塊 B' 切為數個 2×2 的小區塊，得到 b_0'、b_1'、b_2'、b_3'。

步驟 3：針對每個小區塊做量化，得到 $Q_L(b_i')$ 及 R' 圖。

步驟 4：計算每個區塊的變異程度。

步驟 5：將區塊依變化程度由大至小做排序。

步驟 6：取出變化程度最大的前四名所對應的小區塊的 LSBs，形成驗證碼。

步驟 7：將整個大區塊做 MD5 及 HMAC 的計算，得到 16 位元的二進制值。

步驟 8：將驗證碼與步驟 7 所取出的 16 個二進制值做比對，倘若兩者相符則表示此小區塊未遭不法人士竄改，可進行步驟 9；倘若兩者不同，則可得知此處已遭竄改，則此區塊 B' 就無法還原回原始的區塊 B。

步驟 9：將變異數的第五名至第十六名所對應的小區塊做 Lossless G-LSB 的還原，並將取出的值還原至經變異數計算後的前四名所對應的 LSBs。

延續前面的例子，偽裝區塊在經量化後，會與最初藏入資訊前的 $Q_L(b_i)$相等，因此可以計算出與發送端在嵌入資訊前相同的變異數名次。取出變異數前四名，並依序取出它們的 LSBs，與整個區塊 B' 再做 MD5 與 HMAC 比對，若比對不符，表示此區塊 B' 已遭破壞，若比相對相符，則表示此 B' 和發送端的無異，可以進行區塊還原。圖十四展示的是上例中變異數第五名的偽裝小區塊。

128	92
48	94

圖 十四、變異數第五名的偽裝小區塊

我們將它經過量化後得圖十五(a)及圖十五(b)的差異表。

125	90
45	90

(a)量化

(b)差異表

3	2
3	4

圖 十五、量化後小區塊及差異表

依圖十五(b)找出對應的區間，如圖十六。其中差異表的第一個值為 3，故從第三個區間 R_3 再往下切割，第二個值為 2，再往 R_2 切割，依此類推。最後從圖十六的第四層中任意選出一個值為 $(0.7109375\ldots)_{10}$，轉為二進制等於 $(0.\underline{1011}011000010101)_2$。小數點後的 4 個位元即為變異數第一名的小區塊所對應的 LSBs，第 5 至第 16 個位元是變異數第二名至第四名所對應小區塊的 LSBs。我們將這些 LSBs 分別還原到相對應位置，就可還原回發送端原先送出的偽裝區塊。

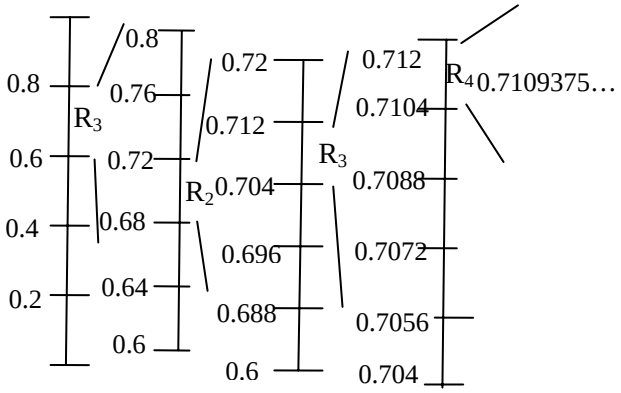

圖 十六、G-LSB 區間圖

4. 預期成果

(1)針對每個小區塊藏入的地方做安排，將資訊嵌入在區塊變異數較大的地方，因此本研究可以找出最適當的藏入地點，改善 L-LAW 法的 PSNR 值。

(2)可達到 L-LAW 法的目的，即偵測此浮水印圖片是否遭不法人士竄改。

5. 結論

本篇論文針對接收端所收到的影像做偵測竄改與否，告知接收端某些區塊已被竄改，而某些未被竄改的區塊進行影像回復的動作，除了可保護發送端的所有權之外，另外也將秘密資訊嵌入於其中。比 Celik M.U. 等學者所提的 L-LAW 法更有效利用區塊中複雜的地方，使影像在網路上傳輸時不容易被不法第三者得知裡頭藏有秘密資訊。

參考文獻

[1]陳俊宏，***採用浮水印技術之影像驗證***，朝陽科技大學碩士論文，2003。

[2]張真誠、黃國峰、陳同孝，***電子影像技術***，旗標出版股份有限公司，頁碼 233-236，2003。

[3]謝仲凱，***應用於影像驗證及回復的階層式浮水印機制***，國立中興大學碩士論文，2004。

[4]戴顯權，***資料壓縮***，紳藍出版社，頁碼 5-28~5-34，2002。

[5]A. Menezes, P. van Oorchot, and S. Vanstone, ***Handbook of Applied Cryptography***, Boca Raton, FL: CRC, 1997.

[6]I. H. Witten, M. Radford, and J. G. Cleary, "Arithmetic coding for data compression," ***Communication on ACM***, Vol. 30, No. 6, pp. 520-540, 1987.

[7]J. Tian, "Reversible data embedding using a difference expansion," ***IEEE Transactions on Circuits and Systems for Video Technology***, Vol. 13, No. 8, 2003.

[8]M. Bellare, R. Canetti, and H Krawczyk, "Keying hash functions for message authentication," ***Proceedings of Cryptology 96***, Vol. 1109, 1996.

[9]M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," ***IEEE Transactions on Image Processing***, Vol. 14, No. 2, pp. 253-266, 2005.

[10]M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Localized lossless authentication watermark (LAW)," ***Proceedings of The International Society for Optical Engineering,*** Vol. 5020, No. 1, 2003. pp. 689-698.

[11]M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: a new framework and an implementation," ***IEEE Transactions on Image Processing***, Vol. 15, No. 4, pp. 1042-1049, 2006.

[12]M. U. Celik, G. Sharma, E. Saber, and A. M. Tekalp, "Hierarchical watermarking for secure image authentication with localization," ***IEEE Transactions on Image Processing***, Vol. 11, No. 6, pp. 585-595, 2002.

[13]R. Rivest, "The MD5 message-digest algorithm," ***IETF Network Working Group***, RFC 1321, 1992.

[14]]The Variance formula can be found at http://www.stat.nctu.edu.tw/MISG/SUmmer_Course/C_language/Matlab/ch4_2_2.htm.