使用大區塊 DCT 係數之影像浮水印方法

An Image Watermarking Method Using Large-block DCT Coefficients

古鴻炎 (Hung-Yan Gu)
台灣科大 資工系 副教授
e-mail: guhy@mail.ntust.edu.tw

鄭彥華 (Yen-Hua Zheng)
台灣科大 資工所 碩士
e-mail: M9315906@mail.ntust.edu.tw

摘要

本論文提出一種強健式、不需原圖即可作萃取的影像浮水印方法，此方法採用大區塊(如 50×50)子影像，把離散餘弦轉換(DCT)後取出的係數，串接成一個序列，當作是一個隨機程序(random process)，再應用該序列的平均值會很靠近 0 的特性，來藏入浮水印位元。實作上先對各個大區塊子影像計算出 DCT 係數，再挑選出中、低頻部分的係數，為了提升強健性，我們另外導入係數位置作變換、和係數值作加權之機制，然後計算跨子影像的 DCT 係數所形成之隨機程序的平均值，再依據給定的門檻值去調整該平均值。經由實驗驗證，本方法在藏入浮水印後仍然可保有相當高的 PSNR 值，並且對於 JPEG 壓縮、改變亮度、剪裁、均化及銳化等攻擊方式，均可表現出良好的防禦能力。

關鍵詞: 影像浮水印、離散餘弦轉換、隨機程序、大區塊子影像

1. 前言

最近關於影像浮水印的研究，藏入的方法大致可分為頻率域上的操作及空間域上的操作；而研究的目標，大體上可分為安全性、隱密性、強韌性、或藏入資料量等方面的改進研究[1]。本論文的研究目標是，在儘量維持影像品質的條件下，去提升所藏入浮水印的強健性。

過去採用離散餘弦轉換(discrete cosine transformation, DCT)係數來藏入浮水印的方法，絕大多數都是先把載體影像切割成 8×8 大小之子影像，然後對各張子影像去計算 8×8 個的 DCT 係數，之後通常就陷入了兩個盲點，第一個盲點是，受限於 64 個 DCT 係數的數量去思考藏入的方法；另一個盲點是，只針對特定的攻擊方式(如 JPEG 壓縮) 來設計藏入方法，而無法抵抗另一種可輕易進行的攻擊方式(如改變亮度)。

因此，本論文就以跳脫前述兩個盲點作為出發點，來研究、設計影像浮水印的藏入方法。為了跳脫 8×8 子影像大小的限制，以求得較大數量的 DCT 係數來思考藏入的方法，我們嘗試把載體影像切割成較大的區塊(如 50×50 之子影像)，然而在大區塊的情況下，一般的觀念仍然是適用的，例如不要修改直流(DC) 附近的極低頻 DCT 係數，以避免發生 blocking effect；不必思考去利用最高頻附近的 DCT 係數，因為那些係數幾乎都會被 JPEG 壓縮所破壞。所以，對於大區塊子影像作 DCT 轉換得到的係數，我們大體上也只能夠挑選中、低頻的 DCT 係數來作利用。

另外，當想要尋找一種可以同時抵抗多種攻擊方式的藏入方法時，我們覺得基於統計特性的藏入方法，是比較能夠達成目標的。因此，我們應用 DCT 係數所形成的隨機程序的平均值會很靠近 0 的特性，來設計一套 DCT 係數的修改流程，以藏入浮水印資料。然後，我們對這個浮水印藏入方法，藉由(a)JPEG 壓縮、(b)改變 Gamma 亮度值、(c)低通濾波、及(d)剪裁等攻擊方式，來作強健性的檢驗，並且用以決定幾個選項的數值。整體來說，我們的方法所藏入的浮水印是不可見(invisible)的，並且作浮水印萃取時，不需用到原始影像，雖然我們的藏入方法的原理並不困難，但是其強健性與影像品質卻是很不錯的，這應是它的特點。

1.1 簡短文獻回顧

過去研究影像浮水印的文獻中，空間域比較常見的藏入方法是，直接改變像素(pixel)的低次位元(bit)來藏入資訊 [2]，空間域方法的好處是，運算處理速度快，但相對的缺點是，當影像遭受攻擊時，較無抵抗能力。至於頻率域上的藏入方法，會先把空間域像素值代入某一種可逆函數去作轉換，以得到頻

率域的係數，然後利用頻率係數的特性(例如人類視覺對於低頻係數較為敏感)，選取適當的係數作修改後以藏入資訊，最常被採用的轉換函數是 DCT 轉換。

Cox 等人提出一種應用展頻技術的影像浮水印方法[3]，從 DCT 係數最低頻的位置開始，挑選出能量較大 1000 個係數，依據浮水印位元序列，修改各對應位置的 DCT 係數以藏入浮水印位元。此方法具有不錯的強健性，不過對影像品質會造成不小的破壞，並且萃取浮水印位元時需要使用到原圖。

過去，使用 8×8 子影像之 DCT 係數來藏入浮水印的研究，例如 Hsu 和 Wu 提出一種將浮水印資訊藏入 16 個 DCT 中頻係數的方法[4]。Luo 等人提出一種結合 JPEG 壓縮的快速且強韌的浮水印技術[5]，利用 4 個相鄰 8×8 區塊之 DCT 係數，來藏入浮水印的一個位元。Zhao 等人提出一種強韌的藏入方法[6]，從每個 8×8 區塊選取一個 DCT 的 AC 係數，來藏入浮水印的一個位元。

李秀月提出一種結合 JPEG 壓縮的 RWJPEG 浮水印法[7]，將浮水印資料藏於 8×8 DCT 計算後的 DC 係數上，並以 zigzag 順序的第一及第二個 AC 係數也用來藏入第二份及第三份浮水印，作為取出浮水印時輔助之用。前述幾篇的研究，很明顯地都受限於 8×8 之子影像大小。

1.2 影像品質量測

對於一張藏入浮水印後的載體影像，其影像品質究竟受到多少影響，一般常用的量測方法是，以 PSNR (peak signal to noise ratio)值來判斷影像品質的變化情形，PSNR 的定義如公式(1) [2, 8]，

$$PSNR = 10 \cdot log_{10} \frac{255 \times 255}{MSE} \tag{1}$$

其中 MSE(mean square error)表示均方差，其值越小則表示與原始影像越相近，對於灰階影像而言， MSE 的定義如公式(2) ，

$$MSE = \frac{1}{MW \times MH} \sum_{i=0}^{MW-1} \sum_{j=0}^{MH-1} (I(i,j) - I'(i,j))^2 \tag{2}$$

其中 MW 與 MH 分別表示載體影像的寬、長的像素個數，$I(i,j)$ 表示在 i, j 位置的影像像素。

至於浮水印本身在藏入前與解出後的相似性的量測，一個常被採用的作法是以正規化關聯值(normalized correlation，NC)來量測解出浮水印的好壞[9]，部分文獻上以相似度(similarity)稱之，其定義如公式(3) ，

$$NC = \frac{\sum_{i=0}^{WW-1} \sum_{j=0}^{WH-1} W(i,j) \cdot W'(i,j)}{\sum_{i=0}^{WW-1} \sum_{j=0}^{WH-1} [W(i,j)]^2} \tag{3}$$

其中 WW 與 WH 分別表示浮水印影像的寬、長的像素個數，$W(i,j)$ 表示在 i, j 位置的浮水印像素。

計算出 NC 值之後，如果 NC=1，則表示浮水印藏入前與解出後是完全相同的，若 $0 \le NC < 1$，則表示浮水印藏入前與解出後有差異。然而 NC 值之計算結果，有時候會有誤判的情形發生[9]，當浮水印的某些像素值，由 0 被改變成 1 時，就會發生無法偵測該像素已被改變的情形發生，因此本論文的改進作法是，將像素值 0 與 1 互換，再去計算一次 NC 值，然後從二次計算出的 NC 值，選取較低的數值作為參考，以求實驗之正確性。

2. 位元藏入之原理與機制

我們基於機率上中央極限定理的特性(當隨機序列的長度愈長，則其 sample mean 愈穩定) [10]，來設計浮水印位元的藏入機制，這個機制可在不需原圖作參考的條件下，萃取出浮水印。我們假設各個子影像作 DCT 後挑選出的 DCT 係數，串接成一長列後會形成一個穩態(stationary)之隨機程序，也就是假設各個取出的 DCT 係數是獨立的、來自同一機率分布 $P(x)$之隨機變數(random variable)。接著，我們可把上述隨機程序依某種分割方式分割成 M 個(浮水印的位元個數)子序列，而各個子序列 $R^m = R_0^m, R_1^m, \cdots, R_{n-1}^m$ 仍然是個隨機程序，然後依據中央極限定理可知，只要子序列長度 n 夠大，則子序列 R^m 的**樣本平均值**(sample mean)會很穩定地趨近機率分布 $P(x)$ 的平均值，亦即樣本平均值的標準差會很小。

雖然前人的研究已經證實，DCT 轉換後求出的係數數值會呈現出平均值為 0 之高斯(Gaussian)分佈[11]，但是為了了解一般影像作大區塊 DCT 後取出的中、低頻 DCT 係數所形成之隨機程序，其**樣本平均值**是否會很接近 0 值，因此我們以 512×512 大小的 Lena 灰

階影像為例，先將影像切割成 50×50 之子影像，再經由離散餘弦轉換，取出 100 個子影像的中低頻部分共 1024×100 個 DCT 係數，這些係數統計出的直方圖(histogram)如圖 1 所示，圖 1 的橫軸數值是 DCT 係數值，我們以 -0.25 至+0.25 為中心區間，然後向左、向右每 0.5 的間隔設定一個區間；另外，縱軸上的數值表示落至各區間的 DCT 係數的個數。對這些 DCT 係數計算它們的的**樣本平均值**，結果得到的數值為**-0.00265**，而樣本標準差則為 **7.134**，所以我們推論一般影像作大區塊 DCT 後，中低頻係數的平均值也會非常靠近 **0**。

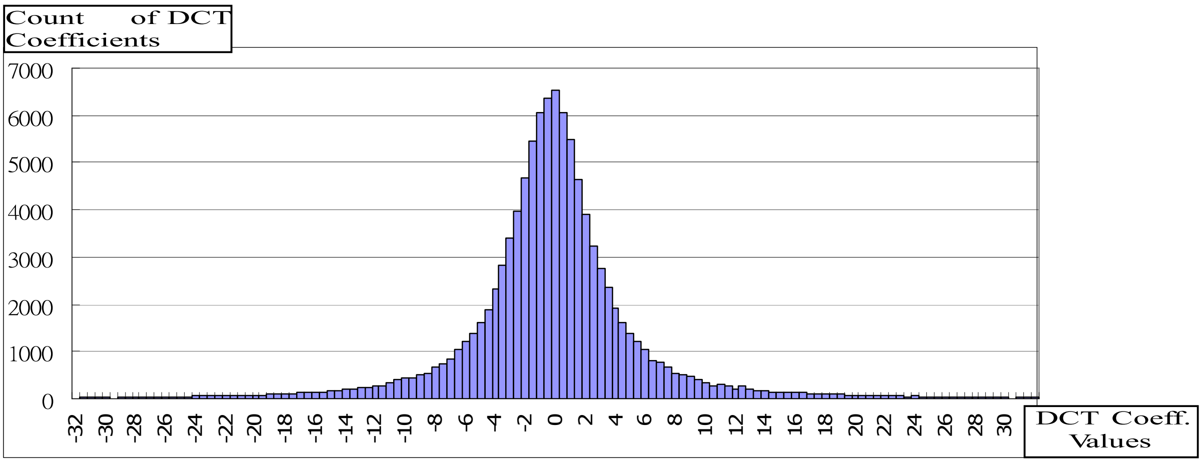


圖 1 DCT 係數數值之直方圖

依據**樣本平均值**會很靠近 0 的特性，我們想到的一個浮水印位元之藏入機制是，當欲藏入的浮水印位元是 1 時，就設法將該位元所對應的 DCT 係數子序列 R^m 的樣本平均值 μ 提升到某一預設的門檻值 *thr* (如 1.0)，這相當於要把子序列裡的各個 DCT 係數 R_k^m 都加上一個差距量(thr - μ)；相反地，當欲藏入的浮水印位元是 0 時，就設法將該位元所對應的 DCT 係數子序列 R^m 的樣本平均值 μ 降低到某一預設的門檻值 $-thr$ (如 -1.0)，這相當於要把子序列裡的各個 DCT 係數 R_k^m 都減去一個差距量($thr + \mu$)。

如此，將來作浮水印位元的解碼處理時，就可直接依據各位元所對應的 DCT 係數子序列 R^m 的樣本平均值 μ 的正負極性，來判斷藏入的位元是 1 或 0，當 μ 大於 0 時，表示藏入的是 1，而當 μ 小於 0 時，就表示藏入的是 0。所以，我們設計的浮水印位元藏入機制，並不需要使用原始的載體影像。

3. 基於大型 DCT 之浮水印方法

本論文提出的浮水印藏入方法，其處理流程如圖 2 所示，分成 7 個處理方塊，分別是(a)子影像切割，(b)大型離散餘弦轉換，(c)係數選取，(d)位置變換與導入加權，(e)位元藏入，(f)反向離散餘弦轉換，(g)子影像組合。以下各子節就對這些處理方塊，作較詳細的說明。

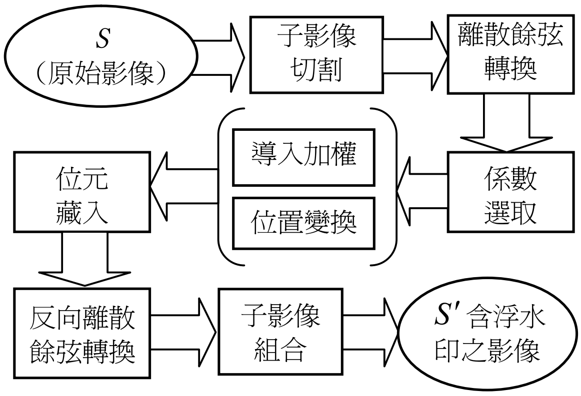


圖 2 浮水印藏入之處裡流程

3.1 子影像切割

許多研究者只將載體影像切割成 8×8 大小之子影像，但是本論文則嘗試切割成較大區塊的子影像。在 512×512 之載體影像的情況，我們測試了四種子影像大小的影響，分別是切割成 46×46，50×50，56×56 及 64×64 等不同的區塊大小，當把載體影像切割成子影像後，設可得到 n 個子影像 S_k，$k = 0, \ldots, n-1$，則 n 的值會分別是 121, 100, 81, 和 64。至於在影像右邊及底部、無法符合區塊大小之區域，則維持原樣，不加以利用。

在實際進行數項攻擊實驗的檢驗後，我們發現切割成 50×50 的子影像是比較好的選擇，因此以下就只考慮 50×50 之子影像大小。雖然切割成 46×46 之子影像，可讓各個隨機子序列的長度變長為 121 個元素，而增進**樣本平均值**的穩定性，但是 46×46 子影像算出的 DCT 係數數值，其防禦能力相對地則是變弱了。

3.2 離散餘弦轉換

一個子影像 S_k 在空間域的像素資料，令其為 $I(x, y)$，$0 \le x, y < N$，在此以如下公式[12],

$$A(i,j)=\sqrt{\frac{2}{N}}\times\sqrt{\frac{2}{N}}C(i)\cdot C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}I(x,y)\cos\left[\frac{(2x+1)\cdot i\pi}{2N}\right]\cos\left[\frac{(2y+1)\cdot j\pi}{2N}\right]$$

$$\begin{cases} C(i),C(j)=\frac{1}{\sqrt{2}} & \text{if} \quad i,j=0 \\ C(i),C(j)=1 & \text{if} \quad i,j\neq 0 \end{cases} \tag{4}$$

作離散餘弦轉換，其中 N 是子影像的長與寬的像素個數。所得到的頻域 DCT 係數在此以符號 $A_{i,j}^{k}$ 表示，k 表示子影像編號，而 i, j 分別表示縱向與橫向的空間頻率索引(index)。第一個(即 i=0 且 j=0)係數最重要，稱為直流係數(DC)，除了直流係數之外，其它的係數，都稱為交流(AC)係數。在實際進行離散餘弦轉換時，我們採用了一種較快速的演算法[13]，可將計算量從 $O(N^4)$減少到 $O(N^3)$。

3.3 係數選取

由於影像的 DCT 係數具有能量集中於低頻的特性，因此我們依據 zigzag 順序(8×8 區塊的例子如圖 3 所示)來排列，將二維的 DCT 係數 $A_{i,j}^{k}$ 轉換成一維的係數序列 B_z^k ，z=0, 1, 2, …，如此 z 值小的，其係數值 B_z^k 通常會較大，且重要性也較大。

接者，我們依照 B_z^k 係數的排列順序，選取中低頻的係數區段，以供後續步驟作浮水印位元的藏入。例如大小為 50×50 的子影像，一種選取方式是，令 L 表示起始位置，並且設定其值為 L = 50×50 / 2 - 3×M / 4, 然後選取出 zigzag 順序 L 之後的 M 個 B_z^k 係數，也就是令選出的係數為 C_z^k ，而且令 $C_z^k = B_{z+L}^k$ ，z=0, 1, 2, …, M-1。在此 M 表示浮水印的位元個數，由於實際測試時使用的浮水印大小是 32×32，所以 M=1024，此情況下 L 不會是負值。

DC	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

圖 3 zigzag 之順序

3.4 位元藏入

第 2 節裡提到，以第 m 個隨機子序列 R^m = R_0^m, R_1^m, ⋯, R_{n-1}^m ，來藏入第 m 個浮水印位元。在此，一種子序列的形成方式是，令子序列的元素 R_i^m 為 C_m^i ，也就是把來自不同子影像的 C_m^0, C_m^1, …, C_m^{n-1} 收集成為一個子序列。

接著，以下式計算出隨機子序列的平均值，

$$Y_m = \frac{1}{n}\sum_{k=0}^{n-1} C_m^k \tag{5}$$

然後，依據第 m 個浮水印位元 W_m 的值，將門檻值 thr 或-thr 與 Y_m 作比較，以決定出調整量 d_m，也就是:

$$d_m = \begin{cases} thr - Y_m, & \text{if } W_m=1 \text{ and } Y_m < thr \\ -thr - Y_m, & \text{if } W_m=0 \text{ and } Y_m > -thr \end{cases} \tag{6}$$

若公式(5)裡的條件不成立，就直接將 d_m 設為 0 值，也就是不需調整。求得調整量 d_m 後，再依下式調整相對應的 DCT 係數，

$$\bar{B}_{m+L}^k = \bar{C}_m^k = C_m^k + d_m, \; k=0, 1, ..., n\text{-}1 \tag{7}$$

如此，隨機子序列的新平均值 $\bar{Y}_m$ 就會變成大於等於 thr 或小於等於-thr。

3.5 位置變換

前一子節裡所採取的隨機子序列的形成方式，優點是簡單，但是缺點是，不同的浮水印位元各自使用專屬的 zigzag 順序編號之 DCT 係數，這會造成很大的防禦能力的差異，其原因在 3.3 子節已提到，就是 zigzag 順序編號小的 DCT 係數，係數值通常是較大，如此當進行 JPEG 壓縮攻擊時，係數值比較不會被轉成 0 值。

因此，我們研究另一種子序列的形成方式，稱為"位置變換"，以讓不同的浮水印位元可以享有約略相同的防禦能力。作法是，在不同的子影像裡，持續以蛙跳方式來決定 DCT 係數的編號，如此就可以均勻地選取到 zig-zag 順序編號小的和大的 DCT 係數，也就是令子序列 R^m 的第 i 個元素 R_i^m 為 $C_{v(i,m)}^i$，而位置變換函數 $v(i,m)$ 的定義為

$$v(i,m) = (i \times F + m) \bmod M \tag{8}$$

其中 F 表示蛙跳的距離，它必須是一個與 M 互質的整數，才不會導致和其它子序列衝撞到同一個 DCT 係數，在本論文裡，我們直覺地設定 F 為 701，而 M 如前述為 1024。

3.6 導入加權

以公式(5)計算隨機子序列的平均值 Y_m 時，及以公式(7)作 DCT 係數調整時，來自各個子影像的 DCT 係數原先都分配給它們相同的加權值，即 $1 / n$。但是當進行"位置變換"的處理之後，一個隨機子序列的元素將會是來自於不同 zigzag 順序編號的 DCT 係數，而不同 zigzag 順序編號的 DCT 係數，很明顯地具有不同的防禦能力。

因此，我們對於不同 zigzag 順序的 DCT 係數導入不同的加權值設定，目前採取的設定方式是一種線性、反比的方式，就是令 DCT 係數 C_j^k 的加權值 α_j^k 為 $1 / (1+j)$。如此，第 m 個浮水印位元對應的隨機子序列平均值的計算方式就要由公式(4)修改成

$$Y_m = \frac{1}{\beta_m} \sum_{k=0}^{n\text{-}1} \alpha_{v(k,m)}^k \cdot C_{v(k,m)}^k \qquad \beta_m = \sum_{k=0}^{n\text{-}1} \alpha_{v(k,m)}^k \tag{9}$$

而且第 m 個浮水印位元所對應的 n 個 DCT 係數的調整公式，也要由公式(6) 修改成為

$$\bar{B}_{v(k,m)+L}^k = \bar{C}_{v(k,m)}^k = C_{v(k,m)}^k + n \cdot d_m \cdot \alpha_{v(k,m)}^k / \beta_m \qquad k=0, 1, ..., n\text{-}1 \tag{10}$$

3.7 反向離散餘弦轉換

作完全部浮水印位元的藏入處理後，第 k 個子影像的第 j 個 DCT 係數 C_j^k，就會被更改成 $\bar{C}_j^k$，也就是 B_{j+L}^k 會被更改成 $\bar{B}_{j+L}^k$。在此先把一維的係數序列 B_z^k，以修改過的 $\bar{B}_{j+L}^k$ 作覆蓋，再重排回成二維 DCT 係數的形式，然後以如下公式[12],

$$I(x,y) = \sqrt{\frac{2}{N}} \times \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)A(i,j) cos\left[\frac{(2x+1)\cdot i\pi}{2N}\right] cos\left[\frac{(2y+1)\cdot j\pi}{2N}\right] \tag{11}$$

作反向離散餘弦轉換，就可以得到空間域、已藏入浮水印的第 k 個子影像。接著將各個轉換回空間域的子影像作組合，就可得到整張已藏入浮水印的影像。實際進行反向離散餘弦轉換時，我們也是採用較快速的、複雜度為 $O(N^3)$ 的演算法[13]。

4. 未受攻擊之藏入實驗

在此以實驗方式探討**門檻值** *thr* 之大小、及有無使用"**位置變換**"和"**導入加權**"的影響，也就是觀察它們對於 PSNR 和 NC 值的影響，一般而言，當 *thr* 值越小，藏入浮水印後所量得的 PSNR 值會較高，但是相對的，取出浮水印後量得之 NC 值也會較低。

關於門檻值的設定，我們分別測試了 0.05、0.1、0.2、0.5、1.0、2.0、3.0、4.0、5.0、7.0 及 10.0 等 11 個值；而"**位置變換**"和"**導入加權**"，則分成三種處理方式，分別是(A)兩者都不使用、(B)只使用**位置變換**、和(C)兩者都使用。在此我們使用如圖 4 所示之 Lena 影像作為載體影像，子影像大小設為 50×50，而浮水印大小是 32×32，並且藏入浮水印後的影像未受到任何的攻擊。

圖 4 灰階 512×512 大小之 Lena 影像

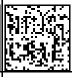
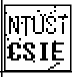
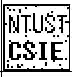
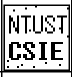
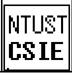
實驗後，量測得到的結果如表 1 所示，由表 1(a)可發現，PSNR 值會隨著門檻值 thr 的增大而減小，這表示 thr 值愈大會造成載體影像愈大的破壞。此外，由表 1(b)可發現，隨著(A)方式、變成(B)方式(只使用位置變換)、再變成(C)方式(也就是位置變換、導入加權兩者都使用)，量得 NC 等於 1 所需的門檻值會愈來愈降低，這表示(B)方式比(A)方式強健，而(C)方式又比(B)方式更為強健。在表 1(b)裡，當量得的 NC 值小於 1 時，我們也將解出的浮水印(原始內容為 NTUST CSIE 等字母)列出，以供參考。

表 1 不同門檻與藏入條件之 PSNR 與 NC 值

(a) 載體影像之 PSNR 值

Thr	0.05	0.1	0.2	0.5	1.0	2.0	3.0	4.0	5.0	7.0	10.0
(A)	51.3	51.2	51.0	50.3	48.7	44.9	42.1	39.6	37.7	34.5	31.4
(B)	51.2	51.1	51.0	50.3	48.6	44.9	42.1	39.8	37.7	34.5	31.4
(C)	50.1	50.0	49.7	48.6	46.8	43.4	40.7	38.1	36.3	33.3	30.6

(b) 解出之浮水印的 *NC* 值

thr	0.05	0.1	0.2	0.5	1.0	2.0	3.0	4.0	5.0	7.0	10.0
(A)	0.741 	0.763 	0.793 	0.969 	1.	1.	1.	1.	1.	1.	1.
(B)	0.944 	0.983 	1.	1.	1.	1.	1.	1.	1.	1.	1.
(C)	0.996 	1.	1.	1.	1.	1.	1.	1.	1.	1.	1.

5. JPEG 壓縮攻擊實驗

5.1 壓縮深度之實驗

對於已藏入浮水印之 Lena 影像，我們作 JPEG 壓縮時，分別以 QF (quality factor) =45、QF=75、和 QF=95 來進行攻擊，以檢驗我們方法對於 JPEG 壓縮的抵抗能力。在此，子影像大小設為 50×50，**位置變換**和**導入加權**兩者都有使用，而對於門檻值 *thr* 的設定，則嘗試前面所列出的 11 種數值。實驗後，我們將結果整理成表 2，由表 2 可知，隨著 JPEG 之 QF 值的減小，門檻 *thr* 值就要調大，才能使 NC 值維持為 1(也就是解出正確無誤之浮水印)，但是當 *thr* 值調大後，藏入浮水印之載體影像的 PSNR 值就會降低，而造成影像品質的衰退。

表 2 不同 QF 時,可使 NC 達到 1 的最低 *thr* 值

壓縮品質	最低 *thr* 值	PSNR	藏入浮水印後之載體影像
QF=45	5	33.51dB	
QF=75	2	36.86dB	
QF=95	0.5	43.56dB	

5.2 與他人之藏入法比較

依據 5.1 子節的實驗結果，本論文方法對於 JPEG 壓縮之攻擊，可說是具有良好的抵抗能力。為了與他人的藏入法比較萃取出的浮水印的品質，在此我們以子影像大小 50×50 及 *thr*=3 之設定來藏入浮水印，然後以 JPEG 壓縮(QF=40、70 及 90)作攻擊，之後再萃取出浮水印。

在三種 QF 值之 JPEG 壓縮攻擊後，本論文方法萃取出的浮水印所量得的 NC 值如表 3 最末一列所示。由表 3 可知，當 QF 值為 70 與 90 時，我們方法所得到的 NC 值都比另三種藏入法的好。雖然 Zhao[6] 與 RWJPEG[7] 之方法可藏入 64×64 較大的的浮水印，不過這兩方法並未測試”亮度攻擊”之項目，並且 RWJPEG 方法使用了 DC 係數，應是無法抵抗

亮度攻擊。此外，Zhao[6]的方法藏入浮水印後，PSNR 值只有 35.13dB，比我們方法的 40.7dB 明顯低很多。

表 3 與 Luo 及 RWJPEG 法的 NC 值比較

藏入法	QF=40	QF=70	QF=90
Luo[5] 32×32 浮水印	**0.856**	**0.909**	**0.932**
Zhao[6] 64×64 浮水印	**0.993**	**0.998**	**0.999**
RWJPEG[7] 64×64 浮水印	**0.873**	**0.965**	**0.991**
本論文方法 32×32 浮水印	**0.904**	**1.**	**1.**

6. 剪裁、亮度、均化、及銳化攻擊

其實在進行各項的攻擊實驗時，除了 Lena 影像之外，我們也測試了其它的影像，如 Boat、Pepper、Baboon 等，實驗結果顯示，我們的藏入方法都可以獲得一致性的效能。在此為了方便，我們就只以 Lena 影像為例，來說明幾項攻擊實驗的進行方式與實驗結果。

6.1 剪裁攻擊之實驗

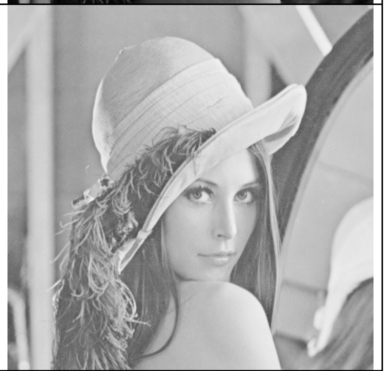
在藏入浮水印後，以剪裁去除載體影像 25% 或 50%之方式作攻擊，之後作浮水印萃取。在剪裁 25%的攻擊方式下，門檻 *thr* 值必需設定到 3 及以上，才能使 NC 值到達 1；而在剪裁 50%的攻擊方式下，門檻 *thr* 值必需設定到 5 及以上，才能使 NC 值到達 1。

6.2 亮度改變之攻擊實驗

Luo[5] 和 RWJPEG[7] 的藏入法，使用 DCT 的 DC 係數來藏入浮水印位元，因此理論上較難抵抗亮度改變的攻擊，然而本論文方法是以 AC 係數來藏入浮水印，所以能夠抵抗亮度改變之攻擊。

這裡分別以不同的 *thr* 值作浮水印藏入，再以 PhotoImpact 軟體設定 gamma 值為 0.5, 2 及 4 作攻擊，之後作浮水印的萃取。我們的實驗結果顯示，在不同的 gamma 值情況，要使 NC 值維持為 1 所需設定的 *thr* 值不完全一致，詳如表 4 所示。

表 4 不同 Gamma 值,可使 NC=1 之最低 *thr* 值

Gamma	*Thr*	
0.5	2	
2	1	
4	1	

6.3 均化攻擊之實驗

這裡分別以不同的 *thr* 值作浮水印藏入，然後對載體影像作均化(smoothing)處理，均化處理之運算元如圖 5(a)所示，為一個 3×3 之遮罩(mask)。均化處理後作浮水印萃取，實驗結果顯示，當 *thr* 值設為 3 時，量得的 NC 值就可達到 0.999，亦即只有 1 個浮水印位元錯誤。

$$\begin{array}{|c|c|c|}\hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \hline \end{array} \qquad \begin{array}{|c|c|c|}\hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array}$$

(a) 均化遮罩　　(b) 銳化遮罩

圖 5 均化與銳化之遮罩

6.4 銳化攻擊之實驗

在此我們分別以不同的 *thr* 值作浮水印藏入，然後對載體影像作銳化(sharpening)處理，銳化處理之運算元如圖 5(b)所示，也是一個 3×3 的遮罩。銳化處理後作浮水印萃取，並且量測 NC 值，結果顯示只要把 *thr* 值設為 1.0 或以上，就能夠讓 NC 值達到 1。因此，本論文方法對於銳化之攻擊，具有良好的防禦能力。

7. 結論

本論文提出了一種強健式、不需原圖即可作萃取的浮水印隱藏方法，它的特點是，原理不困難而卻能具有不錯的強健性與影像品質。此方法把大型子影像區塊作 DCT 轉換所得到的係數，當作隨機程序裡的隨機變數，再應用中央極限定理的特性(當序列的長度愈長，則樣本平均值愈穩定)，來對隨機程序分割出的各個子序列的平均值作調整，以作浮水印位元的隱藏。

此外，在基本的藏入方法之外，我們也對隨機子序列的元素設計了位置變換的機制，以使各浮水印位元，能夠更平均的使用不同頻帶的 DCT 係數；再者，依 zigzag 順序加入係數加權的機制，以使本論文方法對於均化攻擊能有具更好的抵抗能力。經由實驗的結果得知，本論文方法能夠在 JPEG 壓縮、剪裁、亮度改變、均化或銳化等種類的攻擊後，仍然能夠萃取出相當高品質的浮水印，因此可說是具有不錯的防禦能力。

關於 DCT 係數的選取，目前我們只依據一個簡單公式來決定 zigzag 序列的一個編號作為起點(即 3.3 節裡的 *L* 值)；另外關於 DCT 係數的加權，目前我們只是使用線性、反比的加權。未來可再進一步研究此二項議題，以更為提升本論文方法的效能。

參考文獻

[1] 林禎吉、賴溪松，”數位浮水印的技術”，資訊安全通訊第四卷第三期，1998。

[2] 鍾國亮，影像處理與電腦視覺，東華書局，台北，2004。

[3] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, “Secure Spread Spectrum Watermarking for Multimedia”, IEEE Trans. Image Processing, Vol. 6, No. 12, pp. 1673-1687, 1997.

[4] C.-T. Hsu and J.-L. Wu, “Hidden Digital Watermarks in Images”, IEEE Trans. Image Processing, Vol.8, No.1, pp. 58-68, 1999.

[5] W. Luo, G. L. Heileman, and C. E. Pizano, “Fast and Robust Watermarking of JPEG Files”, Fifth IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI.02), 2002.

[6] R.-M. Zhao, H. Lian, H.-W. Pang, B.-N. Hu, "A Watermarking Algorithm by Modifying AC Coefficies in DCT Domain", 2008 International Symposium on Information Science and Engieering, pp. 159-162, 2008.

[7] 李秀月，影像壓縮技術為基礎之資訊隱藏研究，碩士論文，屏東科技大學資訊管理系，2003。

[8] 鍾國亮，資料壓縮的原理與應用，全華科技圖書公司，台北，2004。

[9] 王旭正、柯宏叡，”資訊與網路安全”，ICCL-資訊密碼暨建構實驗室，博碩文化公司，2006。

[10] F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester, A Modern Introduction to Probability and Statistics, Springer, 2005.

[11] R. Reininger and J. Gibson, “Distributions of the Two-Dimensional DCT Coefficients for Images”, IEEE Trans. Communications, Vol. 31, No 6, pp. 835-839, 1983.

[12] Z.-N. Li and M. S. Drew, Fundamentals of Multimedia, Pearson Education, Inc. 2004.

[13] 張真誠、黃國峰、陳同孝，數位影像處理技術，旗標出版公司，台北，2003。