植基於向量量化編碼法的非嵌入式影像版權保護技術

Non-embedded Watermarking Scheme Based on Vector Quantization to Protect the Image Copyright

楊政興 黃樹乾 楊政男 羅坤展
國立屏東教育大學資訊科學所
{chyang, schuang, bm094103, bm095111}@mail.npue.edu.tw

摘要

在本論文，我們提出一個植基於向量量化(Vector Quantization; VQ)編碼法來產生非嵌入式浮水印的影像版權保護技術。我們的方法利用編碼簿，將影像中的區塊與浮水印位元產生關連，並輸出表示關連的 *key stream*，藉由此 *key stream* 來證明影像與浮水印的關連性，達成宣告版權的目的。我們的方法可以彈性調整每個區塊對應浮水印的位元數。與學者 Lin 等所提出的非嵌入式浮水印方法相比較，我們的方法只需要一本編碼簿，而他們的方法需要七本，再者我們的方法不會有某些區塊無法與浮水印產生關連的缺點。最後，我們的方法衍生出強健型浮水印和脆弱型浮水印兩種變型，不但縮減 *key stream* 的長度，而且增加浮水印應用上的彈性。

In the paper, we propose a non-embedded watermarking scheme, which is based on vector quantization (VQ), to protect the image copyright. Our approach applies a codebook to generate a relationship between image blocks and watermark bits, then the relationship is outputted as the key stream. With the key stream, the relation between the image and the watermark is confirmed and the copyright of the image is declared. In our method, the number of bits related to a block is adaptive. Compared with Lin et al.'s approach, our approach need only one codebook; however, their approach need seven codebooks. Moreover, each block can be connected with watermark bits in our approach; however, some blocks can not be connected with watermark bits in their method. Finally, our approach evolves two strategies: robust watermarking and fragile watermarking, which not only to reduce the length of key stream, but also to increase flexibility of application.

關鍵詞：浮水印(Watermarking); 資訊隱藏(Information Hiding); 向量量化(Vector Quantization; VQ); 版權保護(Copyright Protection)

一、簡介

由於數位處理技術發達，要複製或竄改數位媒體是件容易的事，而網路的快速發展，更助長了問題的嚴重性。因此，發展數位媒體之版權保護技術與完整性檢測技術，便成為當務之急。

數位浮水印(Digital Watermarking) [10]

與資訊隱藏(Information Hiding) [13]在資訊安全的領域中，隨著電腦科技的發展以及網路的普及，越來越顯得重要。數位浮水印為保護合法數位媒體的一種技術，其方法通常是將一版權訊號（Copyright Signal）加入合法版權的數位媒體中，媒體的使用者並無法用視覺或聽覺辨別浮水印的存在，也無法輕易移除浮水印。當有版權爭議時，可以取出浮水印來宣告版權。資訊隱藏技術則是將機密資訊藏入數位媒體中，例如將機密訊息嵌入到掩護影像(Cover Image)內，產生偽裝影像(Stego-image)，讓經授權的使用者能從偽裝影像中取出秘密訊息；未授權的使用者則沒有辦法察覺偽裝影像中藏有秘密訊息，進而達到保護訊息的目的。

另一類版權認證的方法是產生非嵌入式浮水印，並可結合數位簽章技術來宣告版權。對影像認證而言，非嵌入式浮水印是從原始影像擷取出特徵值，經過加密處理這些特徵值後作為認證碼儲存。驗證影像時，需先將認證碼進行解密，產生原始影像的特徵值，並且與需要認證的影像所萃取出的特徵值相比較，判斷影像是否被修改過。這一類技術的缺點是需要額外的認證碼，但其好處在於不需要更動原始影像內容。就目前發展來說，針對非嵌入式浮水印機制的研究較少[1, 2, 8]，其中學者Lin 等提出將欲保護的影像與宣告版權用的浮水印緊密結合，利用此產生的 *key stream* 來達成宣告版權的目的。

就影像而言，向量量化(Vector Quantization; VQ)編碼法是一種被廣泛討論的壓縮技術，也有很多學者提出在 VQ 上的浮水印或資訊隱藏技術[6, 7]。本論文針對非嵌入式浮水印方法，提出植基於 VQ 的影像版權保護技術。本論文的結構如下：第二節簡介浮水印與 VQ，以及學者 Lin 等的非嵌入式影像版權保護方法；第三節介紹我們所提出的在 VQ 上之非嵌入式浮水印技術，接下來第四節分析比較我們的技術與學者 Lin 等的技術；最後於第五節提出結論。

二、相關技術介紹

(一) 數位浮水印技術簡介

數位浮水印是一個新興的技術，可以用來追蹤數位資料的傳播與保護智慧財產權。嵌入式數位浮水印技術的概念是嵌入版權資訊到被保護的媒體中，要驗證版權時，只要從媒體中萃取出版權資訊即可。近年來許多探討影像的浮水印技術相繼被提出，若從嵌入的領域而言，可以分為兩大類：第一類為空間域(Spatial Domain)[10]浮水印，另一類為頻率域(Frequency Domain)[5]浮水印。通常，空間域的方法可以嵌入較多的資訊，而頻率域的方法則較具強韌性。頻率域的轉換技術，包括離散餘弦轉換(Discrete Cosine Transform; DCT)[12]、離散小波轉換(Discrete Wavelet Transform; DWT)[11]、傅利葉轉換(Fourier Transform; FT）[3]等。主要是透過這些轉換技術將媒體中重要的部分與不重要的部分區分出來，再將欲嵌入的資料加到所選取的係數中。

在非嵌入式浮水印方面，學者Chang等提出一種不嵌入浮水印於影像中的保護技術[1]，利用頻率域轉換技術將影像中比較重要的部份作為特徵值，然後有一個專門儲存影像特徵值的資料庫，將這些特徵資料與資料庫裡的資料進行比對、過濾查出該影像是否已經註冊，藉此可做為影像是否侵權的判定。學者Chen等[2]提出向量影像保障技術（Vector Image Protection; VIP），他們以VQ 為基礎，改良了VQ 中向量的劃分方法以降低計算上的複雜度，取得受保護影像的重要區域的特徵值儲存

在特徵資料庫中，若懷疑某張影像時，便可以將該影像做轉換後與特徵資料庫裡的資料比對，就可以知道該影像是否未經授權使用。而學者Lin等[8]提出利用編碼簿，將影像中的區塊與浮水印位元產生關連，並輸出表示關連的*key stream*來證明影像與浮水印的關連性，達成宣告版權的用途。

(二) 向量量化壓縮法(VQ)

向量量化(Vector Quantization; VQ)編碼法[4, 6, 7]是一個有效率的失真影像壓縮技術，在失真率有一定水準的情況下，提供低位元率且高影像品質的壓縮方法，加上它的演算法簡單而直覺，所以常被相關領域的論文所引用。

在 VQ 的方法中包含了編碼簿的設計、壓縮及解壓縮程序。首先，將影像分割成大小為 $n \times n$ 的不重疊影像區塊，編碼簿中編碼字的大小亦為 $n \times n$。編碼簿的產生方式，最具代表性的就是 LBG 演算法。在壓縮的過程中，需在編碼簿找出與影像區塊最相近的編碼字。對影像區塊 B 而言，其編碼方式為計算 B 與所有編碼字的歐基里得距離，找出距離最近的編碼字 c，利用編碼字 c 的索引值 I 來編碼。依序找出所有影像區塊最相近的編碼字，並利用其索引值來編碼，即完成了壓縮的動作。解壓縮時可以利用這些索引值去取出編碼簿中的編碼字，然後將其以影像區塊的形式輸出，最後就可以完成影像的解壓縮。

(三) 學者 Lin 等的影像版權保護方法

2006 年學者 Lin 等[8]提出一個非嵌入式的浮水印技術，用於影像版權的保護，其方法是產生 *key stream*，將欲保護的影像與宣告版權用的浮水印緊密結合，利用此產生的 *key stream* 來達成宣告版權的目的。*key stream* 的產生，則是利用七本標號(Label)為 0 到 6 的編碼簿，將每個區塊分別至此七本編碼簿中，找出最接近的編碼字，然後將編碼字的索引值取 2^t 的模數，若所得到的結果符合欲關連之 t 位元的浮水印，則輸出最先符合的編碼簿之標號作為該區塊的 *key* 值；若七本編碼簿都不符合，則輸出$(111)_2$，代表此區塊不與浮水印產生關連。持續上述動作直到所有區塊處理完成。圖 1 為產生 *key stream* 的流程圖。

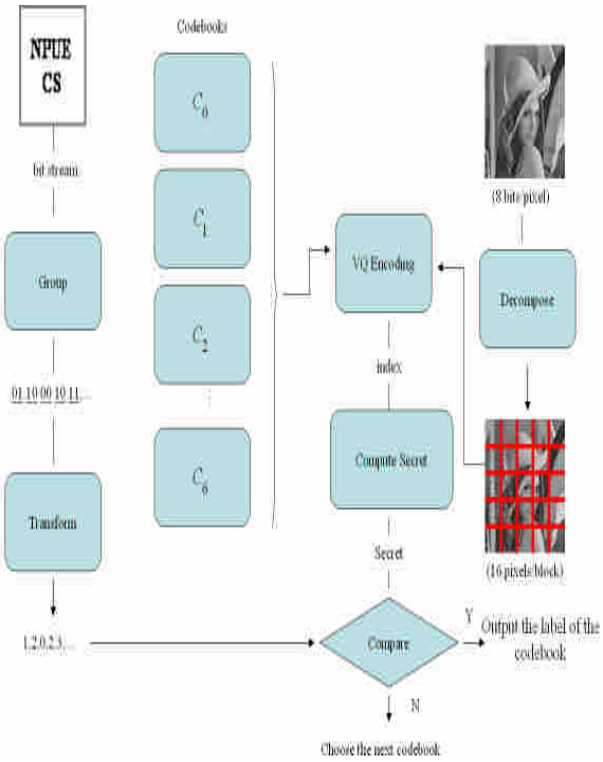

圖 1 學者 Lin 等的非嵌入式浮水印方法產生 *key stream* 之流程圖

假設每個區塊將與浮水印中的 t 個位元產生關連，則其產生 *key stream* 的詳細步驟如下：

Step1: 針對目前正在處理的區塊，到編碼簿 C_0 中尋找最接近的編碼字，假設此編碼字的索引值為 *Index*，利用公式(1)算出 *Secret* 值，該 *Secret* 值定義如下:

$$Secret = Index \bmod 2^t \quad (1)$$

Step2: 如果*Secret*值與*t*位元的浮水印之數值相符，則輸出此編碼簿的標號，否則跳到步驟3。

Step3: 對下一本編碼簿執行步驟 1 和步驟 2，並且重複直到七本編碼簿都被執行過。如果所有的編碼簿所產生的 *Secret* 值，都與 *t* 位元的浮水印不相符，則輸出$(111)_2$表示此區塊沒有對應的浮水印。

Step4: 對所有未處理的區塊，重複執行步驟 1~步驟 3，直至處理完影像的所有區塊。

舉例來說，表 1 為區塊B_1、B_2和B_3，分別從七本編碼簿中找出與其最近之編碼字的索引值。例如：在編碼簿C_0中，區塊B_1找到的索引值為 100；在編碼簿C_1中，區塊B_1找到的索引值為 102。假設浮水印為$(000111)_2$，且 t = 2，則將浮水印位元兩兩分為一組：00 01 11，用十進位來表示分別為 0, 1, 3，利用公式(1)所算得在不同的編碼簿$C_i(i=0,1,2,...,6)$中所得到的 *Secret* 值顯示於表 2。例如：區塊B_1在編碼簿C_0中產生的 *Secret* 值為 0，在編碼簿C_1產生的 *Secret* 值為 2。接著，依序將浮水印所代表的十進位數值，與區塊相連結來產生 *key stream*。第一個浮水印的十進位數值為 0，由表 2 知，區塊B_1所產生的 *Secret* 中，最先符合 *Secret* 為 0 的編碼簿為C_0，所以輸出C_0的標號$(000)_2$，當作是區塊B_1的 *key* 值。接著，下一個浮水印數值為 1，因為表 2 中區塊B_2所產生的 *Secret* 值都不是 1，所以區塊B_2沒有與浮水印生關連，因此產生標號$(111)_2$的 *key* 值，且浮水印數值 1 必須在下一個區塊中繼續處理。因為區塊B_3所產生的 *Secret* 中，最先符合 *Secret* 為 1 的編碼簿為C_2，所以區塊B_3所產生的 *key* 值為$(010)_2$。依此方式，對剩下的浮水印數值和區塊作處理，直到浮水印或整張影像的區塊處理完畢。上述做法可以得知，每個區塊會產生 3 位元的 *key* 值，但是有些區塊無法與浮水印產生關連，例如表 1 和表 2 中的區塊B_2，就沒有和浮水印產生關連。

表 1 區塊B_1、B_2和B_3，利用傳統 VQ 編碼法與不同的編碼簿$C_i(i=0,1,2,...,6)$進行編碼

Codebooks / Blocks	C_0	C_1	C_2	C_3	C_4	C_5	C_6
B_1	100	102	89	68	102	110	130
B_2	120	122	83	124	26	55	36
B_3	78	95	57	70	92	100	105

表 2 區塊B_1、B_2和B_3，在不同的編碼簿$C_i(i=0,1,2,...,6)$中所得到的 *Secret* 值

Codebooks / Blocks	C_0	C_1	C_2	C_3	C_4	C_5	C_6
B_1	0	2	1	0	2	2	2
B_2	0	2	3	0	2	3	0
B_3	2	3	1	2	0	0	1

三、我們的影像版權保護方法

(一) 我們的非嵌入式浮水印技術

我們提出一個全新的非嵌入式影像版權保護技術，我們的方法也是植基於 VQ 上。在我們的方法中，只需用到一本編碼簿，而且不會有某些區塊無法與浮水印產生關連的缺點。圖 2 為我們的非嵌入式浮水印產生 *key stream* 的流程圖，若每個區塊要與浮水印的 *t* 個位元產生關連，對區塊H_i而言，到編碼簿中找出與H_i最接近的2^t個編碼字，依接近程度由近而遠依序命名為c_0，c_1，…，c_{2^t-1}，而其對應的索引值為I_0，I_1，…，I_{2^t-1}，若區塊H_i所對應的 *t*

位元浮水印 W_i 其數值為 x，$0 \le x \le 2^t - 1$，則區塊 H_i 所產生的 *key* 值 k_i 為 I_x。

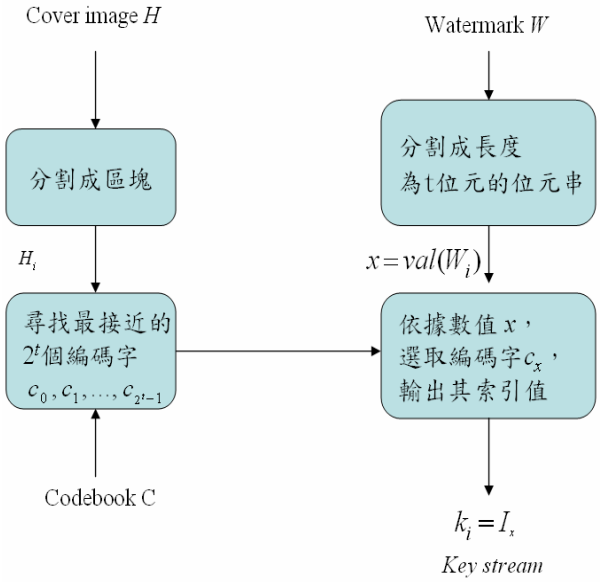

圖 2 我們的非嵌入式浮水印方法產生 *key stream* 之流程圖

舉例來說，表 3 為區塊 H_1、H_2 和 H_3，分別從編碼簿中找出與其最近之 2^t 個編碼字，然後依接近程度由近而遠依序標示為 c_0，c_1，c_2，c_3 之索引值。假設浮水印為 $(000111)_2$，且 $t = 2$，則將浮水印位元兩兩分為一組：00 01 11，用十進位來表示分別為 0, 1, 3，以浮水印所代表的十進位數值，與區塊相連結來產生 *key stream*。以區塊 H_1 為例，H_1 對應到的浮水印數值為 0，因此由表 3 知 H_1 會輸出編碼字 c_0 的索引值 96 來當作 *key* 值。接著，下一個浮水印數值為 1，因此區塊 H_2 會輸出編碼字 c_1 的索引值 118 來當作 *key* 值；第三個浮水印數值為 3，因此區塊 H_3 會輸出編碼字 c_3 的索引值 67 來當作 *key* 值。依此方式，對剩下的浮水印數值和區塊作處理，直到浮水印或整張影像的區塊處理完畢。若編碼簿的大小為 128，由上述做法可以得知，每個區塊都可用 $\log_2 128 = 7$ 個位元的 *key* 值來與 *t* 位元的浮水印產生關連。

表 3 區塊 H_1、H_2 和 H_3，利用我們的 VQ 編碼法進行編碼

Codewords / Blocks	C_0	C_1	C_2	C_3
H_1	96	98	85	64
H_2	116	118	79	120
H_3	74	91	53	67

圖 3 為我們的非嵌入式浮水印的版權驗證流程圖。版權驗證的過程，利用原先產生的 *key stream*，一一取出 *key* 值 k_i，然後把 k_i 當作編碼簿的索引值。另外，假設與 k_i 產生關連的區塊 H_i，其最接近的 2^t 個編碼字依序為 c_0，c_1，…，c_{2^t-1}，則判斷 k_i 是屬於 2^t 個編碼字中的哪一個，假設為 c_j，則輸出 *t* 位元的浮水印 $W_i = j$。

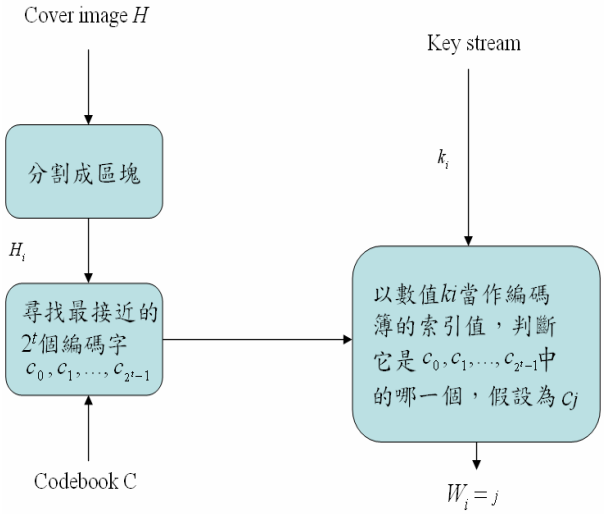

圖 3 我們的非嵌入式浮水印方法的版權驗證流程圖

(二) 我們的強健型與脆弱型浮水印技術

一般浮水印技術可以依據其抵抗破壞的能力，而將它們分成強健型浮水印 (Robust Watermark) 與脆弱型浮水印 (Fragile Watermark)，而這兩類技術也因為

其特性上的不同，分別有著不同的應用。強健型的浮水印技術最大特徵就是對於惡意的破壞有較高的抵抗能力；脆弱型浮水印技術則是用來驗證可疑影像的完整性與真確性。我們透過這種分類方式，將上一節提出的方法做進一步的延伸，發展出強健型浮水印和脆弱型浮水印。

A. 強健型浮水印

假設每個區塊要與 $t = 2$ 位元的浮水印產生關連，既然在我們的方法中每次只找出四個與區塊最接近的編碼字，因此我們特別設計大小為 4 的編碼簿。其產生 *key stream* 的步驟類似，對每個區塊 H_i 而言，依照 H_i 與編碼字間的歐幾里得距離，由近而遠將編碼字依序命名為 c_0，c_1，c_2，c_3，而其對應的索引值分別為 I_0，I_1，I_2，I_3，若區塊 H_i 所對應的 2 位元浮水印其數值為 x，則輸出索引值 I_x 作為區塊 H_i 的 *key* 值。因為編碼簿大小為 4，所以每個 *key* 值只用 2 個位元來表示，比上一節我們原先的方法還要節省，而且，因為編碼簿中四個編碼字之間的距離非常大，因此浮水印可以形成強健型的效果。

B. 脆弱型浮水印

原先我們所提出的非嵌入式浮水印方法，對 $t = 2$ 而言，每個區塊 H_i 到編碼簿中找出最近的四個編碼字，由近而遠依序命名為 c_0，c_1，c_2，c_3，其對應的索引值分別為 I_0，I_1，I_2，I_3，如圖 4 所示。若區塊 H_i 所對應的 2 位元浮水印其數值為 x，則輸出 I_x 作為作區塊 H_i 的 *key* 值，假設原始編碼簿大小為 128，I_x 需要 7 個位元來表示。現在，我們針對索引值 I_0，I_1，I_2，I_3，由小到大給予簡碼 00,01,10,11，如圖 4 所示。如此，我們不再輸出 I_x 當作 *key* 值，而是輸出 I_x 對應的簡碼。例如，若 $x = 3$，則輸出 I_3 所對應的簡碼 10。在此方法中，因為簡碼只需要用 2 位元來表示，所以 key 值亦只需 2 個位元，比 3.1 節中的原先方法還要節省。而且，因為編碼字 c_0，c_1，c_2，c_3 之間的距離較近，所以浮水印可以形成脆弱型的效果。

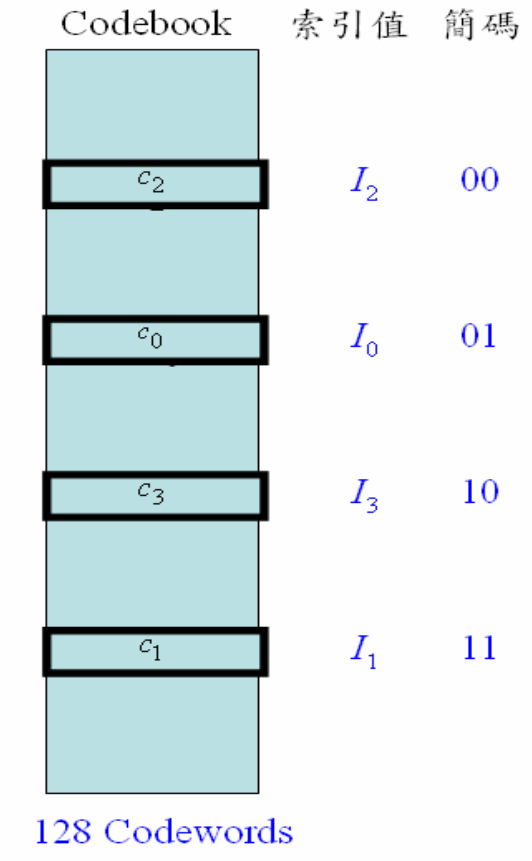

圖 4 我們的脆弱型浮水印之簡碼

四、分析比較

(一) 方法比較

本節我們分析比較我們的方法與學者 Lin 等的方法[8]，並就以下幾個方面來說明我們的優點。

A. 編碼簿大小與數量

我們的方法只需要一本編碼簿，而學者 Lin 等的方法卻需要七本編碼簿。而且，我們的方法甚至可以使用大小為 4 的編碼簿(當 $t = 2$ 時)。所以我們所需要的編碼簿空間小很多。

B. *key stream* 的長度

我們的方法所產生的 *key* 值，其位元數可視 *t* 值的大小調整，當 $t = 1, 2, 3$ 時，*key* 值的位元數分別為 1, 2, 3。而學者 Lin 等的方法，其 *key* 值的長度固定為 3。就一般浮水印的大小來說，$t = 1$ 或 2 是非常足夠的。所以我們的方法可以產生較短的 *key stream*。

C. 浮水印與區塊間的關連性

在我們的方法中，每個區塊都可以和 *t* 位元的浮水印產生關連，不會有失敗的情形。但是在學者 Lin 等的方法中，由於需 mod 2^t 後的餘數與欲關連的 *t* 位元浮水印相等時，才能產生關連，所以在機率均等之下，若 $t = 1$，則任一本編碼簿失敗的機率為二分之一，所以七本編碼簿都失敗的機率為: $\left(\frac{1}{2}\right)^7 = 0.00731$，換言之有 0.7%的區塊無法產生關連；同理，在 $t = 2$ 的情況下，找到的索引值須取 mod 4 的餘數，所以七本編碼簿都失敗的機率為: $\left(\frac{3}{4}\right)^7 = 0.13348$，也就是說有 13%的區塊是無法產生關連；在 $t = 3$ 的情況下，找到的索引值要取 mod 8 的餘數，所以失敗機率為 $\left(\frac{7}{8}\right)^7 = 0.39269$，表示有 39%的區塊無法產生關連。

D. 強弱浮水印的彈性

我們產生 *key stream* 的非嵌入式浮水印技術，衍生出強健型和脆弱型兩種策略，但是學者 Lin 等的方法沒有此種彈性。

(二) 實驗分析

在本節，我們用NC值來判斷取出浮水印之強韌性，其值介於0和1之間。圖5顯示我們所使用的掩護影像Lena，其為大小512 × 512 的灰階影像，圖 6 顯示浮水印 NPUE_CS，其為大小64 × 64的二元影像。在我們方法中所使用的編碼簿為利用Lena圖作為訓練對象，使用LBG演算法來產生大小分別為128個編碼字與4個編碼字的編碼簿，且實驗中每一個區塊大小為4 × 4個 pixel。另外，模糊化和銳利化的攻擊方法，直接採用Adobe Photoshop 7.0所提供的功能。

圖 5 掩護影像 Lena 是大小 512 × 512 的灰階影像

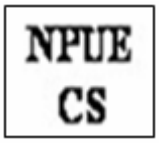

圖 6 浮水印 NPUE_CS 是大小 64 × 64 的二元影像

在表 4 中我們列出了我們提出的強健型和脆弱型兩種策略，在 $t = 2$ 的情況下，分別與浮水印產生 *key stream*，然後原圖受到模糊化、銳利化攻擊後，經由版權驗證流程所取出的浮水印之 NC 值。由 NC 值的大小可以驗證，強健型浮水印的 NC 值明顯大於脆弱型浮水印的 NC 值。

表 4 Codebook 大小分別為 128 和 4 經過模糊化、銳利化後所測得之 NC 值

Codebook 大小	模糊化	銳利化
128 (脆弱型浮水印)	0.732666	0.758943
4 (強健型浮水印)	0.998535	0.994873

五、結論

由於數位多媒體和網路科技的發展快速，使得數位電子資料傳播快速，也因為網路的傳輸過程中，缺乏私人的空間，使得私人的數位影像被大量快速的使用，因此數位影像的版權問題日漸嚴重。近年來，在著作權法制定後，版權問題受到相當的重視。本文所提出的非嵌入式浮水印技術可應用於數位產品(尤其是不可失真數位產品)版權的認證及偵測產品是否遭受更改。我們提出的方法與原先學者 Lin 等所提出的非嵌入式浮水印方法相比較，我們的方法只需要一本編碼簿來證明影像與浮水印的關連性，並且可以彈性調整每個區塊對應浮水印的位元數達成宣告版權的目的。

致謝

本研究接受國科會之計畫編號：NSC 96-2221-E-153-001 和 TWISC@NCKU 計畫編號：NSC 96-2219-E-006-009 的部份經費補助。

參考文獻

[1] C.C. Chang and H.C. Wu, “A copyright protection scheme of images based on visual cryptography,” The Imaging Science Journal, pp. 141-150, 2001.

[2] T.S. Chen and H.R. Wu, "The vector image protection techniques based on vector quantization," in Proceedings of 11th Information Security Conference , 2001.

[3] J.W. Cooley and J.W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” Mathematics Computation, Vol. 19, pp. 297-301, 1965.

[4] R.M. Gray, “Vector quantization,” IEEE Acoustics. Speech and Signal Processing Magazine, Vol. 1, pp. 4-29, 1984.

[5] C. T. Hsu and J. L. Wu, “Hidden digital watermarks in images,” IEEE Trans. Image Processing, Vol. 8, No. 1, pp. 58-68, 1999.

[6] Y.C. Hu, “Grey-level image hiding scheme based on vector quantization” IEE Electronics Letters, Vol.39, No.2, pp.202-203, 2003.

[7] M. Jo and H.D. Kim, “A digital image watermarking scheme based on vector quantization,” Transactions Information system, Vol. E85–D, No. 6, pp. 1054-1056, 2002.

[8] C.C. Lin, Y.C. Hu, and C.C. Chang, “A novel image ownership protection scheme based on rehashing concept and vector quantization,” Fundamenta Informaticae, Vol. 71, pp. 443-451, 2006.

[9] S. Li and W. Li, “Shape adaptive discrete wavelet transforms for arbitrarily shaped visual object coding,” IEEE Transactions on

Circuits and Systems for Video Technology, Vol. 10, pp. 725-742, 2000.

[10] N. Nikolaidis and I. Pitas, “Robust image watermarking in the spatial domain,” Signal Processing, Vol. 66, pp.385-403 , 1998.

[11] A. Piva, R. Caldelli, and A. De Rosa, “A DWT-based object watermarking system for MPEG-4 video streams,” IEEE International Conference on Image Processing, Vol. 3, pp. 5-8, 2000.

[12] C.F. Wu and W.S. Hsieh, “Digital Watermarking Using Zerotree of DCT,” IEEE Trans. on Consumer Electronics, Vol. 46, pp. 87-94, 2000.

[13] Y.H. Yu, C.C. Chang, and Y.C. Hu, “Hiding secret data in images via predictive coding,” Pattern Recognition, Vol. 38, pp. 691-705, 2005.