

應用於 MPEG-4 FGS 智財權保護之具有提升壓縮率之新浮水印技術

王春清
國立彰化師範大學電機工程系

陳柏佑
國立彰化師範大學電機工程系

蕭偉權
力新國際科技

ccwang@cc.ncue.edu.tw

bogy0411@yahoo.com.tw

dick_hsiao@hc.newsoft.com.tw

摘要

隨著網際網路的盛行，多媒體資料在網路上互相傳遞的頻率日漸增高，在網路上觀看這些多媒體資料時，也隨之引發許多版權相關問題，本篇論文將針對多媒體傳播的版權相關問題進行研究。本論文的主要目標將以具有可調式串流的視訊編碼技術為基礎，研究浮水印的版權認證技術。MPEG-4 的 FGS 技術 (fine-granular scalability) 因其具有因應網路頻寬變動而調節串流速率之功能，能提供良好的視訊品質。在本論文中，我們的構想是發展脆弱型的浮水印並將之加入 FGS 的 bitplane 中，藉由脆弱型浮水印萃取的結果來偵測多媒體的完整性。我們的作法是以 bitplane 中一個 64 bits 的區塊(block)為單位，每一個 block 對應二元浮水印的一個 bit 值，而後計算 block 中 bit 值為 1 的 bit 個數，再利用我們提出的位元平面編碼浮水印演算法(Bitplane-coding Watermarking, BCW) 演算法將 block 中一個 bit 的值由 1 改為 0，使 1 總數的奇偶值能夠符合對應的浮水印，在萃取方面則只需判斷 block 中 bit 值為 1 的 bit 總數，即可由奇數或是偶數來決定要取出的浮水印 bit 值是 1 或是 0。而 block 中更改 bit 的位置，我們則是利用了 run-length 的編碼步驟，選擇出較佳的位置，使得編碼時的資料量減少。我們的演算法的特色除了不需要複雜的運算外，更重要的是藉由 bitplane 中加入浮水印的位置選擇，減少了編碼時的資料量，使得視訊壓縮率更進一步的提升。

關鍵詞：MPEG-4、FGS、數位浮水印、視訊串流

1. 前言

近年來，由於網際網路多媒體傳播發展的熱絡，人們漸漸改變過去從平面媒體或電視媒體中獲取多媒體資訊，網際網路已形成各種資訊大量流通的管道。資訊科技的快速發展，電子數位產品不斷的出現，使得多媒體的傳播不再侷限於時間性與地區性，再加上透過網際網路無遠弗屆的便利性，使得人們可以自主性的選擇於隨時隨地下載或是觀賞我們所想要的多媒體資訊。人們容易廣泛的取得多媒體資料，將會引起許多相關違法的問題，特別

是現今的法規對網路所引發的問題尚未有一個完整而詳細的規定，數位複製的技術又使得所有的多媒體資訊能大量複製，如此則容易導致使用者有意無意中去觸犯到智慧財產權擁有者的權利。隨著多媒體資訊的發展，版權問題也相對的倍受重視，因為它可以保障擁有者與創作者的權益。因此如何在利用網路的便利性與效益性的同時，對多媒體智慧財產權(intellectual property right, IPR)做有效且即時的保護，將是刻不容緩的重要議題。

數位浮水印[1,3-5]為保護合法多媒體的一種技術，主要是防止這些數位媒體在網路上遭人下載非法使用，我們可以把合法的版權擁有者(copyright owner)的圖騰(logo)(例如：公司商標、作者名稱、具代表性的圖像...等)加入多媒體內容之中。因為數位多媒體具有可失真的特性，因此能夠容許我們在不嚴重破壞多媒體外觀的情況下加入浮水印來做為版權的識別。基本上，浮水印可分為二大類，一種是可見的浮水印技術(visible watermarking)，例如紙鈔上的浮水印，可由外觀輕易地做出識別，不需要經過解碼萃取等複雜步驟，但是其缺點就是會影響多媒體資訊的美觀，並且容易被利用訊號處理等技術將浮水印加以去除。第二種就是不可見的浮水印技術(invisible watermarking)，也就是將浮水印經演算法藏入多媒體之中，無法由外觀直接看出，再經由演算法萃取出浮水印，用以驗證版權的所有者。圖 1 為不可見浮水印技術的架構圖。

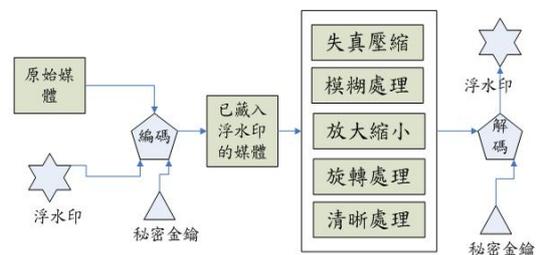


圖 1. 數位浮水印技術架構圖。

數位浮水印做為驗證版權的依據，首先必須要有幾個必要的要求，必須要具有足夠的強韌性(robustness)、要能夠抵抗種種訊號處理(signal processing)的破壞(包括影像壓縮、切割、取樣等)，

並且在加入浮水印後能夠不嚴重破壞影像的品質與原貌。以不可見浮水印的嵌入演算法而言，大致上又分為兩類方法：

第一類屬於空間領域浮水印技術(spatial domain watermarking)，它是直接改變原始數位資料值來達到浮水印之藏入的方法，這種方法具有快速運算的優點，但是它的缺點是不夠堅實，難以抵抗各種訊號處理的破壞。另一種則是屬於頻譜領域浮水印技術 (frequency domain watermarking)，也就是先將原始數位資料轉換至頻譜領域，再藉由改變經轉換後所得的係數值來藏入浮水印的技術，再將具有浮水印的數位資料還原為空間領域的原始多媒體資料。這樣的方法在原始多媒體品質上可能會差了一點，但是能夠讓浮水印對於各種訊號處理的破壞有較高的抵抗性。例如於影像區塊(block)透過轉換成頻譜領域後將產生不同的頻譜(spectrum)分佈，分別為低頻帶區、中頻帶區及高頻帶區，如果選擇在低頻帶區加入浮水印，則浮水印會更為堅實，但是將會嚴重破壞影像的品質，而在高頻帶區加入浮水印，能夠得到較好的影像品質，但是浮水印將顯得不夠堅實，因此在過去大多數浮水印技術所選擇的是將浮水印加在中頻帶區。近年來在這方面的論文也有許多著眼於利用小波轉換的方式來將數位浮水印藏入多媒體資訊中。

由於數位視訊串流(video streaming)在網際網路(internet)與無線網路(wireless network)上傳輸的應用越來越廣泛，對於視訊編碼的目標，以下兩點將被考量:1) 在不同頻寬的情況下，提供良好的品質服務 2) 封包遺失或錯誤的發生。

MPEG-4[2] 標準中細微可調適編碼技術 (fine-granular scalable coding)，它將影像序列壓縮的資料串流分成兩部分：一為基礎層之資料串流 (base layer bitstream)，另一則屬於加強層 (enhancement layer bitstream)，基礎層原始影像與重建影像之間的差值，將以位元平面(bit plane)的技術來形成加強層之資料串流。FGS(fine-granular scalable coding)[6-8]使得加強層的資料串流可隨著通道頻寬之變化任意的捨棄(arbitrarily truncate)，因此 FGS 的優點如下：

- 提供相當好的頻寬適應性 (bandwidth adaptation)。
- 具有錯誤回復(error recovery)的性質。

FGS 編碼的串流具有提供連續位元率與連續品質的特性，因此我們選擇將數位浮水印版權保護技術建立在 FGS 視訊編碼技術的基礎之上。圖 2 與圖 3 為 FGS 編碼器與解碼器的主要架構圖。

MPEG-4 FGS 的架構需要 2 個編碼器，一個為基礎層的編碼器，另一個則為加強層編碼器。在基礎層中，編碼器利用視訊在時間上的相關性來進行 motion-compensation 的編碼，而在加強層的編碼器中，其編碼器輸入的訊號是原始影像與重建後影像差值，再經過 DCT 轉換的結果。換句話說 FGS 編碼器的輸入就是 DCT domain 下的量化誤差

(quantization error)。而編碼的過程則是將輸入訊號經過 Shift Bitplane function、Find Maximum function、最後再經由 Bitplane VLC function 的編碼而產生 enhancement layer 的串流。每一個 DCT FGS 剩餘值碼框組成了 N_{BP} 位元平面 (bitplane)，而後 FGS 加強層編碼器掃描剩餘值信號從最顯著的位元平面(most significant bitplane)到最不顯著的位元平面(least significant bitplane)。每一個 bitplane 是由 16×16 互不重疊的巨方塊(macroblock)所構成，每一個 macroblock 包含 4 個 8×8 luminance(Y)block 和 2 個 chroma block(U 和 V)。再經由 Run-length 編碼過程形成可變長度的編碼，構成了 FGS 壓縮的位元串流。而在接收端，FGS 位元串流藉著一個可變長度解碼器(VLD)被解碼。

FGS 編碼的另一大特色就是能夠透過 Selective Enhancement 與 Frequency Weighting 來對位元平面 (bitplane) 做適應性量化 (adaptive quantization) 的調整。

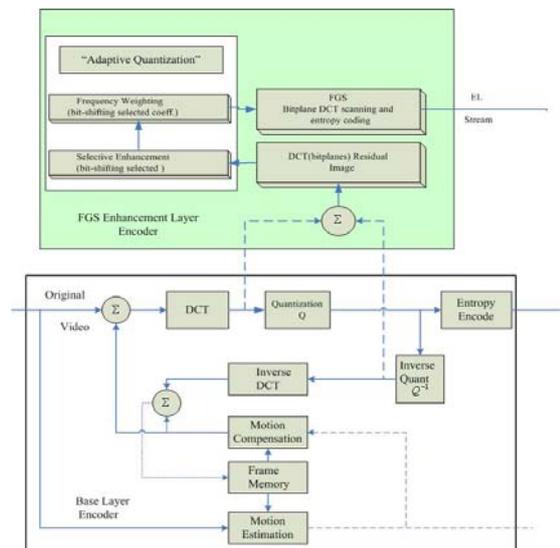


圖 2. FGS 編碼端架構圖。

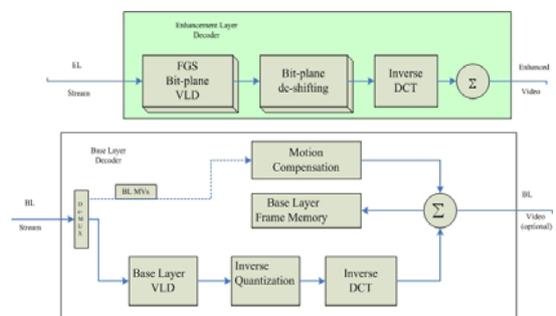


圖 3. FGS 解碼端架構圖。

Selective Enhancement 和 Frequency Weighting 是可以同時被運用的，這兩個工具都是在加強層內改變 DCT 係數的掃描順序，因此促使有興趣的區域或是低頻的 DCT 係數較早被編碼、較早被傳送、也

較有機會被解碼端所接收到。Selective Enhancement 是在 FGS enhancement layer 上的位元平面 (bitplane) 選定 macroblock 再利用 up-shifting 的方式將所選定的 macroblock 加以強調，達到凸顯重要的 macroblock 的效果。而 Frequency Weighting 則是針對 bitplane 的 DCT 係數值利用 shifting 的方式產生凸顯的效果，通常是將低頻區的係數值加以 up-shifting。

我們將在第二部份描述加入浮水印的構想與方案，第三部份是我們提出的位元平面編碼浮水印演算法，第四部份是實驗結果，在第五部份做分析與討論，最後在第六部份做個結論。

2. 加入浮水印的構想與方案

在此，我們希望在 FGS 的加強層中加入一個脆弱型的浮水印，作為驗證多媒體完整性的依據。由於脆弱型浮水印對於各種攻擊的抵抗力是非常低的，因此只要對多媒體稍加攻擊，或是改變多媒體內容，皆會造成萃取出來的浮水印無法達到百分之百的完整性。如此一來，我們即可由萃取出來的浮水印的完整性與否來判斷此一多媒體是否經過非法的竄改或是更動。而本論文的重點為如何在 FGS 的編碼方式中加入浮水印，並且藉由此位元平面編碼浮水印演算法 (Bitplane-coding Watermarking, BCW) 達到提高壓縮率的效果。針對提高壓縮率方面，我們將加入浮水印的每個區塊 (block) 有選擇性的提高 run-length 的效率，用以減少編碼時的資料量來達成提高壓縮率的目的。另外，我們於加入浮水印的規則上可做相應的調整，例如將浮水印分割成四個部份加入，每一個部份依序重複加入對應的畫面，此規則可以讓我們從萃取出浮水印是否有缺失的部份，來判斷多媒體的畫面總數是否完整，或是由浮水印缺失的部份來判斷缺少了哪些畫面。將浮水印隱藏於加強層的主要理由是：1) 由於僅有加強層的視訊串流時，視訊品質並不甚好，駭客對無加強層之視訊串流並不感興趣。2) 加於加強層對視訊品質的影響較低，相對於基礎層這些由 DCT residual 所組成的加強層視訊串流資訊較無影響整體的視訊品質。因此我們將在計畫中朝此方向發展新的浮水印演算法，以保護視訊串流之智慧財產權。

3. 位元平面編碼浮水印 (Bitplane-coding Watermarking) 演算法

加強層的剩餘值信號被展開成從最顯著的位元平面 (most significant bitplane) 到最不顯著的位元平面 (least significant bitplane)，FGS 加強層編碼器掃描剩餘值信號從最顯著的位元平面 (most significant bitplane) 到最不顯著的位元平面 (least significant bitplane)，因此每一個 bitplane 中所含的值不是 0 就是 1。圖 4 為一加在 bitplane 中加入

浮水印的方法示意圖，我們以 bitplane 中 8×8 (64 bits 長度) 的 block 為一個單位，並將浮水印 S 中每一個 bit 透過演算法加入不同的 block。

以浮水印 S(0,0) 值為例：如果浮水印 S 中 S(0,0)=1，我們即計算所選擇被加入的 block 中 1 的總數，假若總數為奇數則不做任何改變，假若為偶數，則改變 block 中其中一個 bit 使得 1 的總數變為奇數。同理，如果 S(0,0)=0，則所選擇被加入浮水印的 block 中 1 的總數，假若總數為偶數則不做任何改變，假若為奇數，則改變 block 中其中一個 bit 使得 1 的總數變為偶數。在此我們需注意的是某些 all-zero block 的情況，即是 block 的 64 個 bit 的值皆為 0，使得 1 的總數為 0，在此情況下，我們選擇不對此 block 做處理，跳過此 block，繼續重複上述方法加入浮水印。

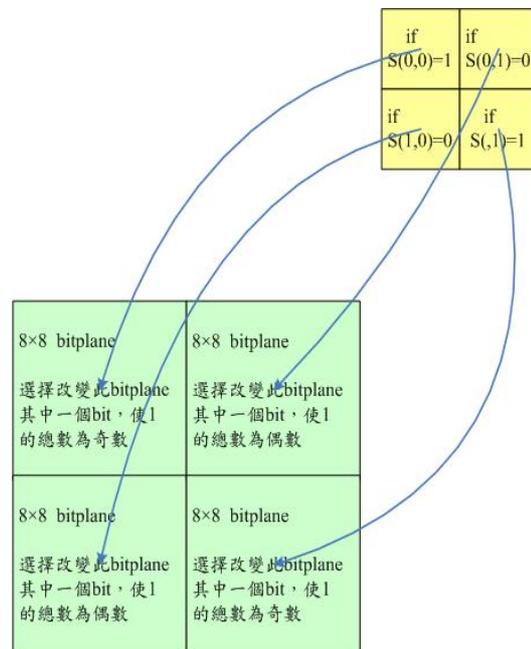


圖 4 本論文所提加在 bitplane 中加入浮水印的方法示意圖。

到底要選擇 block 中那一個 bit 作改變，我們將提出如下的思考策略：

- 選擇適於 run length coding 的 bit 作改變：亦即所選擇的 bit 作改變時將可增進 run-length 的長度，因此，反而增加編碼效率。

FGS 加強層的編碼方式採用了 (RUN, EOP) 的符號來進行編碼，並以一個 block 為基本單位。(RUN, EOP) 中的 RUN 值代表是此組 (RUN, EOP) 所對應的 1 前面有多少個 0，而 EOP 值代表為此組所對應的 1 是否為此 block 中最後一個 1，如果是最後一個 1 則表示此 block 編碼完成並以 EOP=1 表示，反之則為 EOP=0，若是此 block 沒有 1 的存在，即是 64 個 bit 皆為 0，就以特殊編碼 ALL_ZERO 來做處理。由此定義我們可知在一個 block 中除了 ALL_ZERO 的情況外將至少有一組或是一組以上

的(RUN, EOP), 圖 5 是 BCW 演算法藏入浮水印示意圖, 其中圖 5(a)是未加入浮水印前的 block 資

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) 未加入浮水印前的 block

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b) 加入浮水印後的 block

圖 5 BCW 演算法藏入浮水印示意圖。

訊, 我們以圖 5 為例, 根據 zig-zag 的順序, 可得到的 (RUN, EOP)分別(8, 0)、(13, 0)、(18, 0)、(10, 1), 依據上述策略, 我們將未改變 bit 前 block 中所有的(RUN, EOP)組別作一個比較處理, 在此之中找出要更改的 bit 位置, 並且保證此 bit 經過更改之後, 對應此位置的新的(RUN, EOP)中 RUN 的值在同一 block 的所有(RUN, EOP)組別中是最大的, 因此我們在圖 5(a)中選擇更改第 3 個 1, 使得重新計算過後的(RUN, EOP)為以下幾組(8, 0)、(13, 0)、(29, 1), 第三組的 RUN 值為 29, 是所有組別中最大值, 如此一來, 在編碼時, 即能達到最大程度的資料量縮減, 達成我們演算法進一步提高壓縮率的目標, 圖 5(b)是加入浮水印後的 block 資訊。

而在浮水印的萃取方面, 圖 6 為萃取浮水印的示意圖。如果接收端接收到的 bitplane 中 1 的總數(Tc)為奇數, 則可萃取到浮水印的資訊為 1, 如果 bitplane 中 1 的總數(Tc)為偶數則萃取到浮水印的資訊為 0, 若 W_n (n=0,1,...) 代表浮水印的第 n 個資訊則利用下面簡單的方程式可得萃取到浮水印的資訊。

綜合上述方法, 我們將我們所發展的可調適視訊串流浮水印的方法詳細說明於圖 7。而在接收端

$$W_n = \begin{cases} 1 & \text{if } Tc = \text{odd} \\ 0 & \text{if } Tc = \text{even} \end{cases} \quad (1)$$

收到帶有浮水印資訊的 BL Stream 後, 我們首先透過可變長度解碼器(Variable Length Decoder, VLD), 將 BL Stream 進行解碼成為帶有浮水印資訊的 bitplane, 而後利用 Eq.(1)可簡單從中萃取出浮水印, 詳細流程說明於圖 8。

在這個演算法研究當中, 我們發現幾個議題值得探討:

- 如何從 bitplane 中選擇一個位元作為浮水印資訊隱藏的地方?

策略 → 在 bitplane 中選擇一個適合增進 Run Length 編碼的位元。因此可意外贏得位元率的降低。

- 浮水印萃取演算法極其簡單, 僅需計算位元 1 的總數, 不用記憶浮水印資訊的位置。因此極適合應用於可調式視訊串流的傳輸場合作浮水印技術之發展。

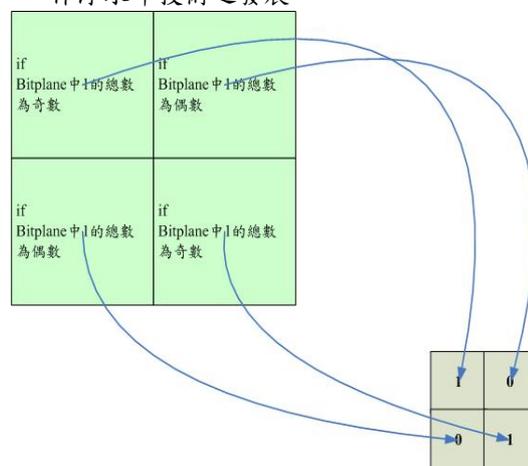


圖 6. 由 bitplane 中萃取浮水印的示意圖。

4. 實驗結果

我們在實驗平台的建構與安排如下, 我們將於 MPEG-4 FGS 編碼器的編碼過程中加入數位浮水印, 其後於解碼過程中萃取出數位浮水印。為了測試數位浮水印對於多媒體的篡改或是更動情形是否能順利的偵測出來, 我們將使用一些破壞處理來對已加入浮水印的視訊串流作攻擊測試。另外, 我們用解碼器將已加入數位浮水印的視訊串流加以解碼, 進行品質的影響實驗, 以評估視訊品質的改變。

我們加入的浮水印為 40x40 像素大小的二元影像, 圖 9 是我們加入的浮水印, 並且採用分割加入浮水印的方式, 將浮水印分成四個部份, 每一個 frame 都依序加入一部分浮水印, 第一個 frame 加入第一部份的浮水印, 第二個 frame 加入第二部份的浮水印, 第三個 frame 加入第三部份的浮水印, 第四個 frame 加入第四部份的浮水印, 接著第五個畫面再重複加入第一個部份的浮水印, 餘此類推,

循環加進所有的 frame 中。因此在萃取上，每一個 frame 都可以取出一部份的浮水印資訊。在未經攻擊時我們依序從”Bus”的視訊片段(video sequence)中萃取出六個 frame 的浮水印列於圖 10。而後我們進行視訊的品質實驗，圖 11. 為”Bus” video sequence 未加入浮水印與加入浮水印的畫面比較上二張圖是未加浮水印的，PSNR 值皆為 45.121dB，而下二張是已加入浮水印的，PSNR 值皆為 39.68dB。在圖 12.我們將未加入浮水印與加入浮水印的”Bus” video sequence 中對前十個 frame 測量其 PSNR 並將結果畫成折線圖比較，縱軸為

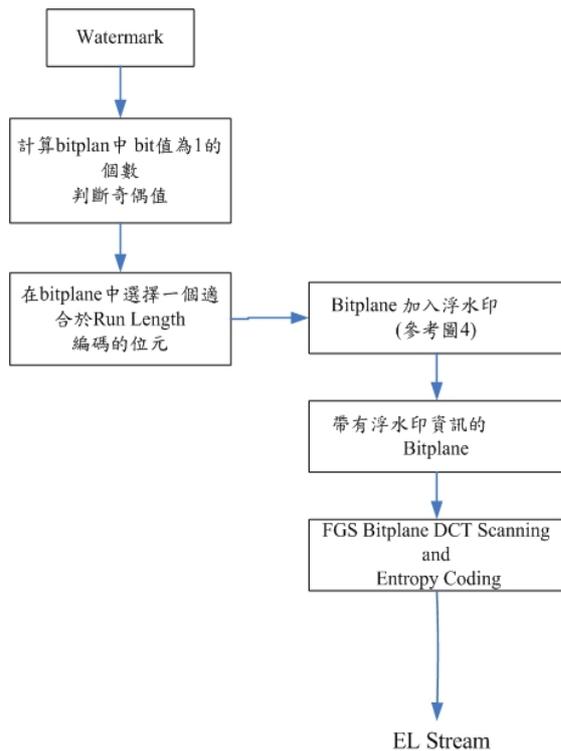


圖 7. 可調式視訊串流浮水印的隱藏方法詳細說明圖。

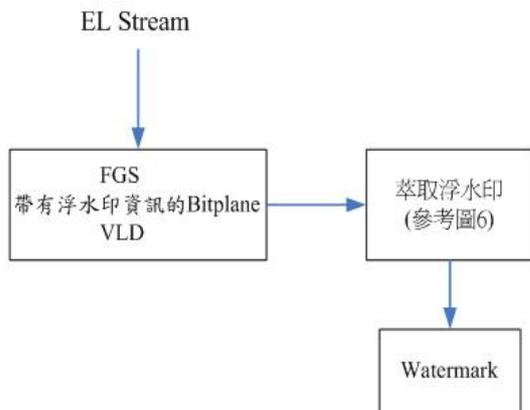


圖 8. 可調式視訊串流浮水印萃取的方法示意圖。
第一個 frame 到第十個 frame，藍色折線代表 PSNR 值單位為 dB，橫軸為 frame 的順序，依序從了以原始視訊為基準與未加入浮水印下編解碼後

所產生的視訊測出的 PSNR 值，粉紅色折線代表



圖 9. 加入的浮水印原圖。

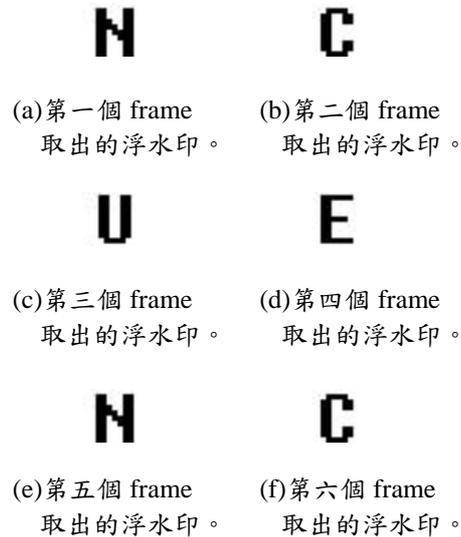


圖 10. “Bus” video sequence 未經攻擊萃取出浮水印

了以相同的原始視訊為基準，加入浮水印編解後的視訊測得的 PSNR 值。圖 13. 為”Mobile” video sequence 未加浮水印與加入浮水印的畫面比較，上二張是未加浮水印的畫面，由左而右 PSNR 值分別為 48.131dB 和 45.121dB，下二張為已加入浮水印的畫面，由左而右 PSNR 值分別為 35.826dB 和 38.588dB。圖 14. 為”Mobile” video sequence 視訊中未加入浮水印與已加入浮水印前十個畫面的 PSNR 折線圖。由以上的視訊品質結果，我們從肉眼上來觀察，其品質並沒有太大的改變，仍在主觀視覺可以接受的範圍，而從 PSNR 的折線圖可看出，加入浮水印的確是會降低些許的 PSNR 值。PSNR 值降低較多的原因在於我們是選擇 bitplane 中的 MSB 層加入浮水印，在此考量的因素是 FGS 在網路傳輸時，會根據頻寬的大小來選擇 bitplane 送出的層數，因 MSB 層是優先被選擇傳送的，較具價值性，因此選擇在 MSB 層加入浮水印。另外在攻擊實驗的安排上，我們是將”Bus” video sequence 加入浮水印之後將編碼完成的檔案抽去其中幾個 frame 的資料再進行浮水印的萃取，圖 16. 為前六個 frame 的萃取結果。從實驗結果可看出，經過抽去 frame 的攻擊測試之後，取出的浮水印有部分的缺失，並非連續的 N、C、U、E，證明了此影片的 frame 總數是不完整的。

我們將未加入浮水印與加入浮水印編碼過後視訊片段的 Enhancement layer 檔案大小與壓縮率

列於表 1，其結果顯示出，我們所提出的演算法的確可以達到壓縮率在進一步提升的效果。

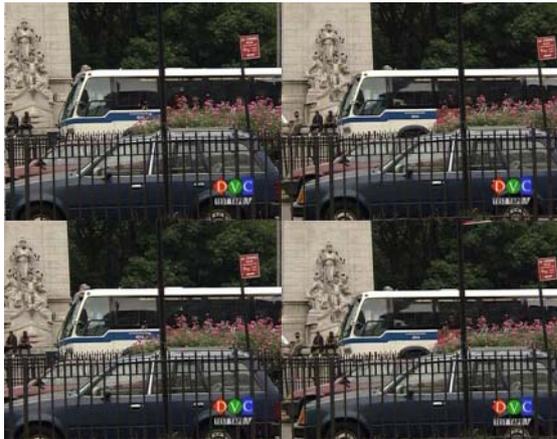


圖 11.上二張為未加浮水印的”Bus” video sequence 畫面，PSNR 值皆為 45.1205dB，下二張為加入浮水印的畫面，由左而右 PSNR 值為 42.1102dB 和 38.1308dB。

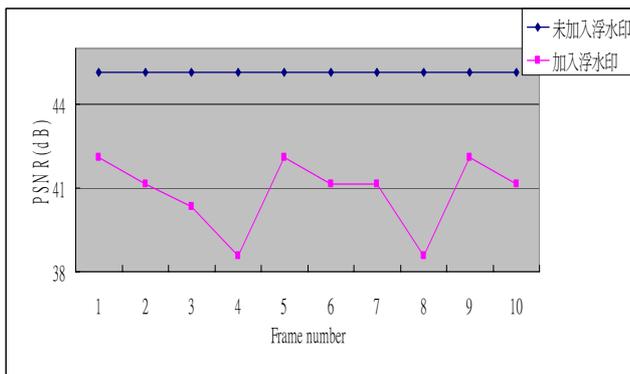


圖 12.”Bus” video sequence 未加入浮水印與加入後的 PSNR 比較圖。



圖 13.上兩張為未加浮水印的”Mobile” video sequence 畫面，PSNR 值皆為 48.1205dB，下二張為已加入浮水印的畫面由左而右 PSNR 值分別為

38.5884dB 和 41.1411dB。

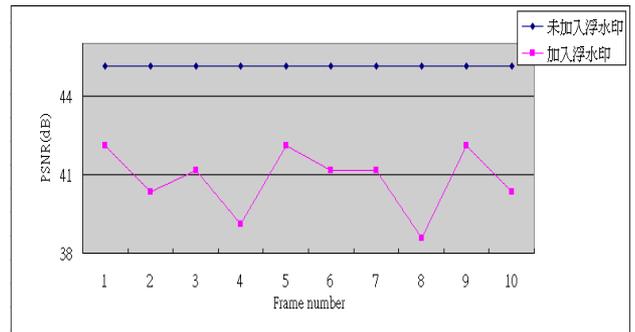


圖 14. “Mobile” video sequence 加入浮水印之後的視訊 PSNR 比較圖。

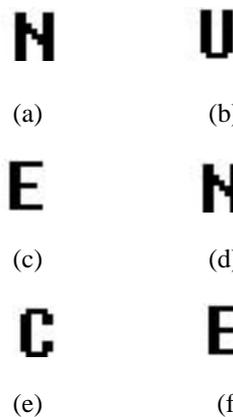


圖 16. (a)~(f)依序是”Bus” video sequence 抽掉其中幾個 frame 之後連續萃取出浮水印。

表 1.各種視訊片段加入浮水印後壓縮率比較表。

檔案名稱	原始視訊大小	嵌入浮水印後視訊大小	壓縮率
Ball	295KB	292KB	1.0 %
Akyio	4238KB	4147KB	1.6 %
Bus	4974KB	4909KB	1.3%
Foreman	8702KB	8587KB	1.2%
Mobile	14395KB	14234 KB	1.1%

5. 分析與討論

本論文所提出的演算法除了有效的提供數位浮水印的藏入視訊的此一基本要求外，最大的特點是我們將編碼時採用的 Run-Length 步驟加以利用。在 bitplane 中選擇出最佳的位置，使得經過 VLC 編碼之後能達到壓縮率的提高。而在加入浮水印的規則上，我們選擇將浮水印分為四個部份，第一部份加入第一個畫面，第二部份加入第二個畫面，第五個畫面則重複加入第一部份的浮水印，餘此類推。藉由此規則，我們即可由萃取出浮水印缺失的部份來推測遺失了哪些畫面。但是隨著浮水印的加入，勢必伴隨著視訊品質的改變，因此如何在達到加入浮水印與壓縮率提高的前提之下使得視訊品質不會受到太大的影響，也是我們所要考量

的一大要素。

6. 結論

本論文所提出的位元平面編碼 (Bitplane-coding Watermarking) 數位浮水印技術，視訊的數位版權的完整性提供一個良好的解決方案，並且由於我們的 BCW 演算法十分簡單，不需要經過複雜的演算即可達成浮水印的藏入與萃取，並且加入浮水印之後對於品質的影響仍在肉眼可以接受的範圍，除此之外還可以達到壓縮率進一步的提升。未來我們將在視訊品質許可範圍內，再使壓縮率更為提高，另外浮水印的應用範圍絕不僅僅侷限於 MPEG-4 FGS 方面，無論是靜態影像或是動態視訊，都有智慧產權這方面的問題，因此，未來我們會更進一步研究靜態影像或是動態視訊的各種壓縮的標準，例如 JPEG2000、H.264 等，期望將我們的演算法加以改良，應用到其他的壓縮標準上，達到更完善廣泛的應用。

誌謝

本研究計畫係由台灣國家科學委員會所協助，特此誌謝(計畫編號 NSC 94-2213-E-018-018)。

參考文獻

- [1] 張真誠,黃國峰,陳同孝 “電子影像技術”，旗標出版有限公司，2003年。
- [2] 戴顯權，陳滢如，王春清 “多媒體通訊原理標準與系統”，紳藍出版社，2002, 2月。
- [3] Xia-Mu Niu; Zhe-Ming Lu; Sheng-He Sun, “Digital watermarking of still images with gray-level digital watermarks”, *IEEE Trans. on Consumer Electronics*, Vol. 46, Issue: 1, Feb. 2000, pp.137 – 145.
- [4] C. Hsu and J. Wu, “Hidden digital watermarks in images,” *IEEE Trans. on Image Processing*, Vol. 8, No. 1, pp.58-68, JANUARY 1999.
- [5] K. Hashida and A. Shiozaki, “A method of embedding robust watermarks into digital color images,” *IEICE Trans. Found.* Vol. E81-A, No.10, pp.2133-2137, 1998.
- [6] Hayder M. Radha, Member, Mihaela van der Schaar, and Yingwei Chen, “The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming Over IP” *IEEE Trans. on Multimedia*, Vol. 3, No. 1, MARCH, 2001, pp. 53-68.
- [7] Weiping Li, “Overview of Fine Granularity Scalability in MPEG-4 Video Standard”, *IEEE*

Transactions on Circuits and Systems for Video Technology, Vol.11, No.3.pp. 332-344, March 2001

- [8] M. van der Schaar, Y. Chen, and H. Radha, “Embedded DCT and wavelet methods for fine granular scalable video: Analysis and comparison,” in *IVCP 2000, Proc. SPIE*, vol. 2974, Jan. 2000, pp. 643–653.