

# 以殘餘誤差為基礎之視訊浮水印技術

劉柏青

國立聯合大學資訊工程學系

U9424008@smail.nuu.edu.tw

蔡丕裕

國立聯合大學資訊工程學系

pytsai@nuu.edu.tw

## 摘要

在視訊浮水印技術中，許多的方法都是將浮水印(Watermark)嵌入單一影格中的影像(I-frame)，也就是將視訊當成影像，然後套用影像浮水印的技術來處理。在本論文中，我們利用視訊影格間的動態向量(Motion vector)與殘餘誤差(Residual error)的特性來對影格間的影像(P-frame)進行浮水印的嵌入，而嵌入的方式也不需要複雜的演算法。實驗結果顯示，此種方法不但能保持視訊的品質，也能容受 MPEG 的壓縮攻擊。

**關鍵詞：**MPEG、視訊浮水印

## Abstract

With the video watermarking, most of watermarking schemes hide the watermark in intra-frame (I-frame). They still apply image watermarking to video watermark embedding. In this paper, we employ motion vector and residual error of inter-frame processing characteristic to insert watermark into inter-frame (P-frame) of video. The proposed watermarking scheme is simple and effective. The experiment results showed that the proposed scheme not only preserves a high video quality but also resists MPEG compression.

**Key word:** MPEG、Video watermarking

## 1、前言

浮水印技術(Watermarking)是一種將浮水印(Watermark)藏入欲保護的資訊中，在外觀上不易被發掘或識破，來達到隱蔽或偽裝欺瞞的效果。透過嵌入浮水印的處理，當欲保護的資訊中出現爭議或質疑時，經由取出嵌入其中的浮水印得以驗證受爭議之資訊。此技術一

方面可提供資訊所有權的一種解決方法，另一方面，浮水印也可以是機密資訊，經由浮水印技術的偽裝效果，為資訊的安全傳送提供另一種管道。所以說浮水印技術也是一種偽裝術。浮水印技術早被前人所應用，而在這個資訊爆炸的時代，數位技術的蓬勃發展，也帶領著浮水印技術更進一步，被人們所研究以及運用。

數位浮水印技術是將欲保護的數位多媒體作為偽裝的對象，也稱為遮罩媒體(Cover Media)，而要藏入的資訊可以是具有驗證所有權標記的個人簽名、公司機關的標誌(Logo)，也可以是完整性驗證的認證碼(Authentication Code)，這些資訊都可通稱為浮水印。嵌入浮水印的多媒體稱為偽裝媒體(Stego Media)。由於嵌入的技術使偽裝媒體不易被人的視覺感官所察覺，所以能夠達到欺瞞的效果。

在視訊浮水印的技術中，許多技術都將浮水印嵌入單一影格中，也就是在影格中的 Intra-frame (I-frame)，因為，I-frame 在視訊編碼中會保留有完整的資訊，所以 I-frame 可以看成是一張影像，當影像的浮水印的技術已經很成熟時，這些技術可以很輕易的轉換到視訊上來。然而，視訊媒體仍有許多影像媒體所沒有的特性。例如相鄰的視訊畫面高度的關聯性，這些特性正是視訊媒體所獨具，而單獨影像媒體所沒有的，因此，視訊浮水印技術應該運用其獨具之特點來發展，才能真正適合其需求。

在目前所研究的視訊浮水印的技術中 Chae 對視訊的 I-frame 先作離散餘弦轉換(DCT)再更改 DCT 係數，以藏入浮水印[1]。Wang 和 Izquierdo 在 MPEG2 中篩選合適的 block 做隱藏並更改 DCT 值[2]，Shang 在 MPEG 中選擇一組相同的 DCT 係數做隱藏，並且能夠還原原始視訊[3]，Hartung 將浮水印隱藏在 I-frame 的 DCT 中[4]，Yi-Chun Liao、Chung-Han Chen、Shih Timothy K.和 Tang

Nick C 共同研究以三維陣列作輔助，利用適合做 LSB 的資料做隱藏[5]，Peng Zheng、Bo Zhao 和 Min-zhong Liu 提出係數演算法的壓縮係數做隱藏[6]，這些方法都是對 I-frame 或是單獨一張影像進行浮水印的嵌入。

但也有少數研究將浮水印嵌入影格間的畫面(P-frame)，如 Lee 選擇合適的動態向量做隱藏[7]，Zhang 選擇將浮水印嵌入在較大的動態向量值[8]，Fang 則對動態向量計算後的角度位置來進行嵌入[9]，Dengpan Ye 提出的改變殘餘誤差的 EQSP(equal quantization step position)來進行隱藏[10]。

在本論文中，我們提出將浮水印嵌入影格間，藉由修改影格間的殘餘誤差來達到嵌入的目的，而且本方法在取出浮水印時不需要使用到原始的視訊資料。

## 2、相關技術

數位視訊在一般的情況下是以 RGB 三原色的數值所構成的影像，再以快速拍照的方式將所拍的影像按時間進行排列，等到播放時再以一張一張影像依照時間貼出，因此一般未經壓縮視訊所需的儲存體是一張一般影像的好幾倍，所以通常都會將錄好的視訊進行壓縮。而本方法是以 MPEG4 壓縮的方式，將視訊容量縮小，並利用 MPEG 殘餘誤差處理的特性，來達到抵擋壓縮攻擊的效果。

由於 MPEG 在編碼之前，必須要先對動態向量做預測，而最好的動態向量預測，就是能讓找到的 macroblock 的誤差越小越好。但事實上，要找到真正無誤差的 macroblock 幾乎不可能，所以儘可能找誤差最小的 macroblock 來做動態向量，基本上有區域搜尋和全域搜尋法，最準確的方法是搜尋一整張畫面，然後，將整張畫面可能適合的 macroblock 都搜尋過，此方法稱作全域搜索法(Full Search Algorithm)。但此演算法較費時，因此，我們以區域搜索法來替代全域搜索法。區域搜索法演算法較簡單，演算時間也較短，但它搜尋到的 macroblock 誤差卻不一定是整張影像最小的。換句話說，就是只能找出區域中最好的 macroblock 來替代，所以它只是區域中最佳的。我們在本論文的實驗中使用區域搜索法來進行，區域搜索法是依照本張影像中的 macroblock，對前一張影像所設定的區域中的每個 macroblock 進行比較。例如，搜尋區域

為 8x8 macroblocks，最中間的 macroblock，其位置欲比對 macroblock。

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

圖 1. Macroblock of 4x4 pixels

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

圖 2. 區域搜尋次序

區域搜尋演算法：

步驟一：假設 macroblock 的大小為 4x4 pixels，如圖 1 所示。

步驟二：假設搜尋區域為 8x8 macroblocks，如圖 2，圖 1 之 macroblock 依序和圖 2 的 64 個 macroblocks 進行比較。

步驟三：找出最小殘餘誤差的 macroblock，將此座標位置和原影像裡 macroblock 的座標位置做相減得到動態向量。

原本的 macroblock 經由區域搜尋後計算出最小殘餘誤差，再將此殘餘誤差經由 DCT 轉換成編碼，和動態向量一起保存起來。影格間藉由儲存這兩參數，來還原整個影像。若 macroblock 在區域搜尋所得之殘餘誤差過

大，而採用自行 DCT 編碼，因此某些 macroblock 是根據此編碼而進行還原，但此方法壓縮量較低且誤差也較大，因此在 P-frame 出現也較少，假如過多就會變成以 I-frame 的方式，直接在影格中進行編碼。

### 3、研究方法

因為大部分的視訊浮水印技術都針對影格中單一影像獨立進行，也就是只對 I-frame 做改變，通常這樣的方式會導致後面的影格變動增加，使得視訊的品質降低。而一般的視訊通常以 P-frame 居多，I-frame 最少，所以，在 I-frame 中能夠嵌入的量也比較少。因此，本方法是嘗試將浮水印嵌入影格間，也就是 P-frame，並且做少部分的更動，期望降低視訊品質的失真，同時能嵌入浮水印的選擇也較多。另外，由於浮水印係嵌入 P-frame，對視訊的壓縮攻擊會有較高的強韌性。本方法包含浮水印的嵌入程序與取出程序。

#### 3.1 嵌入程序

本方法主要是從 P-frame 中找尋殘餘誤差的方式，由搜尋到的動態向量做動態補償後，對殘餘誤差進行修改，而達到隱藏目的。而利用殘餘誤差做嵌入浮水印的構想在於許多的方法都是將誤差忽略，只藉由更改動態向量的方式達到嵌入的目的，此方式可能會讓視訊的品質降低許多，因此我們認為將誤差做些微的修改可以使視訊的品質不會降低太多。因此，我們根據區域搜尋的方法，在原始視訊中，每次以兩張連續的 frames，找出動態向量，並進行動態補償，計算出殘餘誤差。

當殘餘誤差已經計算出來後，就可以進行浮水印的嵌入，假設現在要藏入的浮水印位元為 0，那我們就藉由修改殘餘誤差非 0 的個數為偶數。反之，假如現在要藏入 1，那我們就將殘餘誤差為 0 的數量變成奇數個。這樣的方法類似位元檢核 (parity check) 技術，是一種快速且有效率的方法。圖 3. 為浮水印資訊藏入的流程圖。

依照這樣的方式依續將浮水印藏入。浮水印嵌入演算法如下：

設浮水印長度為  $n$ ，總誤差參數為  $k$ 。

步驟一：對原始視訊以兩張連續的 frames 為一組，對其中的 macroblock 用區域搜尋，找出其最小總誤差的動態向量來進行補償。若總誤差為 0，則將其中一個殘餘誤差改為非 0 的數值，並且在範圍內其他的總誤差不會影響到變為 0，修改後還原成視訊的狀態。進入步驟二。

步驟二：對連續兩張為一組的 frames 用區域搜尋法找出其最小總誤差的動態向量來進行補償。加總 macroblock 的殘餘誤差值，假如總誤差大於  $k$ ，則這個 macroblock 不藏入，否則，進入步驟三。

步驟三：如要藏入 0，修改所有殘餘誤差為零的個數變成偶數。如要藏入的是 1，將修改殘餘誤差為零的個數為奇數。藏完後  $n=n-1$ ，如果  $n \neq 0$  跳回步驟二找下一個 macroblock。若此 frame 以藏完，則找下一組 frame；假如  $n=0$ ，則執行步驟四。

步驟四：將隱藏完的殘餘誤差，依序將視訊經由動態向量的補償後，還原成偽裝視訊。

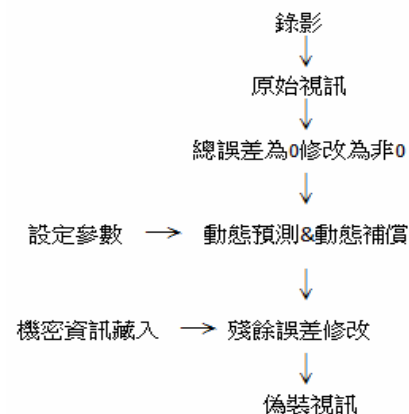


圖 3 資訊藏入流程圖

#### 3.2 取出程序

取出浮水印的程序，類似浮水印嵌入程序，先取出已經藏入有浮水印的視訊，依嵌入程序計算 P-frame 的動態向量與殘餘誤差，假如 macroblock 的殘餘誤差中誤差值為 0 的個數是偶數，那麼就取出浮水印位元 0，假如此

macroblock 的殘餘誤差為 0 的個數是奇數，那麼取出的浮水印就是 1，依此類推，直到將浮水印取完為止。圖 4. 為浮水印取出流程圖。

浮水印取出演算法如下：

浮水印長度為  $n$ ，總誤差參數為  $k$ 。

步驟一：先將偽裝視訊的連續兩張為一組的 frames，對其中的 macroblock 用區域搜尋法找出其最小總誤差的動態向量來進行補償，如總誤差大於  $k$ ，則這個 macroblock 未藏入浮水印，對下一組 macroblock 重新執行步驟一。假如總誤差小於  $k$ ，進入步驟二。

步驟二：假如所有殘餘誤差為零的個數是偶數，則取出浮水印 0；假如所有殘餘誤差為零的個數是奇數，則取出浮水印 1。取出後  $n=n-1$ ，如果  $n \neq 0$  跳回步驟二找下一個 macroblock。若此組 frames 找完，則接著找下一組 frames；

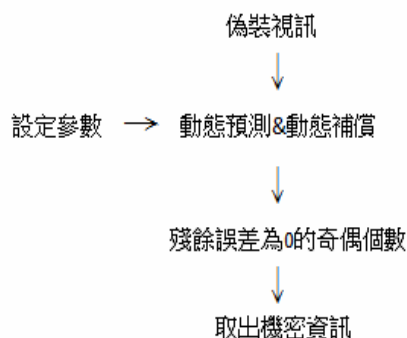


圖4 資訊取出流程圖

### 3.3 問題與討論：

本方法有三個問題須克服，第一點是在視訊殘餘誤差中，假如藏入前殘餘誤差就已經全部都是 0，換句話說，就是完全吻合或說該 macroblock 是靜止狀態，這種情況會造成浮水印取出不正確，所以我們必須在一開始要藏入前，對總殘餘誤差是 0 的 macroblock 做修改，方法為將其中一個是 0 的殘餘誤差更改為非 0 的數，更改時要保持總殘餘誤差不為 0。

第二點是在壓縮中的 P-frame，除了由動態補償的 macroblock 組成外，還有一些是由動態補償後的總殘餘誤差過大，而自行編碼。

因此，對於第二點的解決方法，就是在實驗測量的數據中，設定一個總殘餘誤差的參數，藉由調整參數來提高浮水印的強韌性。

最後一點是考慮畫面中物體移動的速度所導致的問題，如果因為物體的位移量超出了區域搜尋的範圍，使得沒法找到總誤差較低的動態向量，造成此 macroblock 的總誤差超出所設的參數範圍，藏入的資訊量就大幅減少，因此，在本實驗中，我們設定一秒鐘拍攝三十張畫面的方式來防止影像與影像間產生物體移動過快的問題或是將搜尋框加大的方式來處理。

## 4、實驗結果

我們的方法應用在 4 個不同的視訊上，每個視訊大小都是 320\*240，兩秒 60 張影像的視訊，然後分別藏入圖 5 之浮水印，每張浮水印都是 60\*60 灰階圖轉黑白後藏入，而 4 個視訊除了實驗此方法的品質和完整度，還有實驗一些會影響品質高低和完整度，而視訊中移動的物體有手和嘴巴。除此之外，還有實驗物體移動速度和動作。



圖 5 浮水印影像 60x60 bits

從實驗的結果可知，可藏入的容量平均都有 6000 bits 以上，但主要還是要看視訊特性和參數來決定大小；另外，每個嵌入浮水印的視訊都有相當高的 PSNR 值，在未壓縮偽裝視訊平均 PSNR 都有 60 以上，而壓縮過後的 PSNR 也都有平均 36 以上，一般人肉眼比較難辨識出來；在取出後的浮水印都有高完整度，即使經過 MPEG4 壓縮後也有 75%到 90%的完整度，在未壓縮的視訊中，浮水印都能完整的取出；表 1 為視訊的編號 1-4，分別表示不同的視訊；四種不同的浮水印 Nuul、csie、Nuucat、Nuu2；壓縮後完整度依照每種視訊特性的不同而有所差異，但我們藉由調整參數來提高完整度，視訊 4 就是藉由調整參數而提高完整度；而每個視訊可藏入的量也不一定，會依據所捨定的藏入參數大小、視訊本身的特性和動作的快慢來決定；藏入參數調整後會使完整度提高但不一定的調高就會上升，主要看視訊的特



性，可藏入量則是參數越高藏入量越大。表 2 顯示每個視訊都有相當高的 PSNR 值，但會因為藏入參數提高而視訊品質下降；受到壓縮攻擊過後的原始視訊 PSNR 值平均是 35 以上，這代表壓縮後藏入的方法不太會影響壓縮後視訊的品質；而動作的速度會影響取出的完整度，這是因為速度太快殘餘誤差較大比較接近藏入參數但卻沒有達到隱藏要件，而壓縮後會造成殘餘誤差縮小，造成沒有嵌入的部分也變為有嵌入，因此完整度會下降；而因為每個視訊的特性不同，壓縮後原本藏入使用的總誤差參數會有偏移，所以取出時就要設定參數，達到最高的完整度，而通常最好的取出參數在藏入參數的負 1000 之間；而藏入參數也要依據視訊特性去做調整，實驗測出的結果通常藏入參數是在 9000 左右，但也有像視訊編號 4 一樣的特例，就必須提高藏入參數，但參數越高會造成 PSNR 值下降，而藏入參數也會有一個最大值的完整度取出，超過最大值完整度就會下降。

表 1 紀錄機密資訊名稱、視訊編號、壓縮後機密資訊完整度、壓縮後的偽裝視訊大小、藏入設定的參數和視訊可藏入的容量

視訊編號 (320*240)	機密資訊 (60*60)	壓縮後完整度	偽裝視訊大小(kb)	可藏入容量 (bit)	藏入參數
1	csie	90.22	328	8927	9000
2	csie	87.30	320	8400	9000
2	Nuu1	86.94	320	8400	9000
2	Nuucat	87.41	320	8400	9000
2	Nuu2	86.86	320	8400	9000
3	csie	86.38	320	7756	9000
3	Nuu1	87.80	320	7756	9000
3	Nuucat	86.44	320	7756	9000
3	Nuu2	84.30	320	7756	9000
4	csie	76.5	330	6524	9000
4	csie	81.27	330	6759	12000
4	csie	79.94	330	6893	14000
4	csie	79.66	330	7019	16000
4	csie	76.66	330	7151	18000
4	csie	84.38	330	7270	20000
4	Nuu1	86.16	330	7270	20000
4	csie	81.83	330	7843	30000

表 2 紀錄視訊編號、壓縮前後視訊品質(PSNR)、做的動作跟動作速度、藏入和取出時設定的參數

視訊編號	壓縮前 psnr	壓縮後 psnr	動作(速度)	藏入參數	取出參數
1	66.37	36.75	講話(正常)	9000	8770
2	68.15	36.95	動手(慢)	9000	8300
2	68.19	36.95	動手(慢)	9000	8300
2	68.47	36.94	動手(慢)	9000	8300
2	68.18	36.95	動手(慢)	9000	8300
3	63.73	36.02	動手(正常)	9000	8610
3	64.39	36.02	動手(正常)	9000	8610
3	64.22	36.01	動手(正常)	9000	8610
3	64.19	36.02	動手(正常)	9000	8610
4	64.68	35.99	動手(快)	9000	8445
4	61.77	35.98	動手(快)	12000	11550
4	60.63	35.99	動手(快)	14000	13575
4	59.9	35.99	動手(快)	16000	15925
4	59.79	35.99	動手(快)	18000	17610
4	58.78	36.0	動手(快)	20000	19680
4	58.10	35.99	動手(快)	20000	19680
4	54.96	35.99	動手(快)	30000	29740

圖 6 為視訊編號 1 的原始視訊，總共兩秒鐘 60 張影像，說話的動作。



圖 6 編號 1 的原始視訊

圖 7 為視訊編號 1 的偽裝視訊，藏入 CSIE 後的偽裝視訊，能夠取出 100% 的完整度，視訊的品質高。



圖 7 編號 1 的偽裝視訊

圖 8. 是從圖 7 偽裝視訊取出的 CSIE，機密圖的完整度和 PSNR 視訊品質



圖 8 從圖 7 取出的機密圖

圖 9. 是視訊編號 1 經過壓縮過後的偽裝視訊，因為經過 MPEG4 壓縮的關係，視訊品質降低了許多，但肉眼依然看不出來；而受到壓縮攻擊後，機密資訊也無法完整取出。



圖 9 視訊編號 1 壓縮後的偽裝視訊

圖 10. 是經過壓縮攻擊取出的機密資訊，我們利用調整後的參數，取出的完整度有 90%，但因為壓縮的關係 PSNR 值大幅下降。



圖 10 壓縮攻擊取出的機密資訊

圖 11. 為視訊編號 2 的原始視訊，總共兩秒鐘 60 張影像，揮手的動作。



圖 11 視訊編號 2 的原始視訊

圖 12. 為視訊編號 2 的偽裝視訊，藏入 NUU1 後的偽裝視訊，能夠取出 100% 的完整度，視訊的品質高。



圖 12 編號 2 的偽裝視訊

圖 13. 是從圖 12 偽裝視訊取出的 NUU1 的完整度和視訊品質。



圖 13 從圖 12 取出的機密圖

圖 14. 是視訊編號 2 經過壓縮過後的偽裝視訊，因為經過 MPEG4 壓縮的關係，視訊品質降低了許多，但肉眼依然看不出來；而受到壓縮攻擊後，機密資訊也無法完整取出。



圖 14 視訊編號 2 壓縮後的偽裝視訊

圖 15. 是經過壓縮攻擊取出的機密資訊，我們利用調整後的參數，取出的完整度有 86%，但壓縮的關係 PSNR 值大幅下降；因為動作比說話要多，所以完整度會稍微降低。

## 5、結論

在本論文中，我們利用修改殘餘誤差的這種方法，來發展視訊浮水印技術，此方法不需複雜的運算就可以藏入和取出浮水印，由於本方法是隱藏在連續影格間，對一些隱藏於影格中的方法來說，比較能降低壓縮後對視訊的影響，而且能藏入的 frame 也較多。我們的方法

有較大的資訊藏入量，也可以容忍 MPEG4 的壓縮攻擊，同時，也能維持視訊的品質。



圖 15 從圖 14 取出的機密圖

## 參考文獻

- [1] Chae, J.J.; Manjunath, B.S., "Data Hiding in Video," **International Conference on Image Processing, (ICIP 99)**. Vol. 1, pp.311-315, 1999.
- [2] Yulin Wang and Ebroul Izquierdo, "High-Capacity Data Hiding in MPEG-2 Compressed Video," **Congrès Recent trends in multimedia information processing**, pp. 212-218, 2002.
- [3] Yueyun Shang, "A New Invertible Data Hiding In Compressed Videos or Images," **Third International Conference on Natural Computation**, Vol. 5, pp. 576-580, 2007.
- [4] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," **Signal Processing**, Vol. 66, pp. 283-301, 1998.
- [5] Yi-Chun Liao; Chung-Han Chen; Shih, Timothy K.; Tang, Nick C.; "Data Hiding in Video Using Adaptive LSB," **Pervasive Computing (JCP)**. Ppp.185-190, 2009.
- [6] Peng Zheng; Bo Zhao; Min-zhong Liu; "An Efficient Method to Hide Information in MPEG Video Sequences," **International Conference on Multimedia Information Networking and Security, 2009. (MINES '09)**. Vol. 2, pp. 114-117, 2009.
- [7] Hyun-woo Lee; Dong-su Seong; Keon-bae Lee, "Design and Implementation of Efficient Error Concealment Algorithm Using Adaptive Selection of Adjacent Motion Vectors and Data Hiding," **7th International Workshop on Enterprise networking and Computing in Healthcare**

**Industry**, pp.303-306, 2005.

- [8] Jun Zhang; Jiegu Li; Ling Zhang, "Video Watermark Technique in Motion Vector," **Symposium on Computer Graphics and Image Processing**, pp.179-182, 2001.
- [9] Ding-Yu Fang; Long-Wen Chang, "Data Hiding For Digital Video with Phase of

Motion Vector," **IEEE International Symposium on Circuits and Systems**, pp. 51-54, 2006.

- [10] Dengpan Ye; "A new adaptive watermarking for real-time MPEG videos," **Applied Mathematics and Computation**, Vol.185, pp. 907-918, 2007.