

以區塊DCT 係數相似性改良 JPEG/MPEG 編碼品質之研究

¹許志仲, ²張軒庭

光電與資訊實驗室, 國立雲林科技大學電機工程系

{¹g9412716, ²htchang}@yuntech.edu.tw

摘要

本論文提出一種可改善靜態灰階與彩色影像壓縮標準 (Joint Picture Expert Group, JPEG) 之影像編碼技術。每個影像區塊經離散餘弦轉換 (Discrete cosine transform, DCT) 後之係數再進行量化 (Quantization), 許多區塊係數之間可能存在著相似關係。我們尋找與編碼區塊 DCT 係數相似的區塊係數, 對於這些區塊係數只編碼其索引值 (Index value), 並且給予預設之霍夫曼 (Huffman) 碼當成判斷依據。另一方面, 本方法可以直接應用到動態影像壓縮標準 (Motion Pictures Experts Group, MPEG) 中, 不需更改標準編碼器的架構。實驗證明, 本論文所提之方法可以有效提升重建影像品質與提供簡易的影像內容保護機制。

關鍵字: 靜態影像壓縮標準、動態影像壓縮標準、相似區塊、影像壓縮。

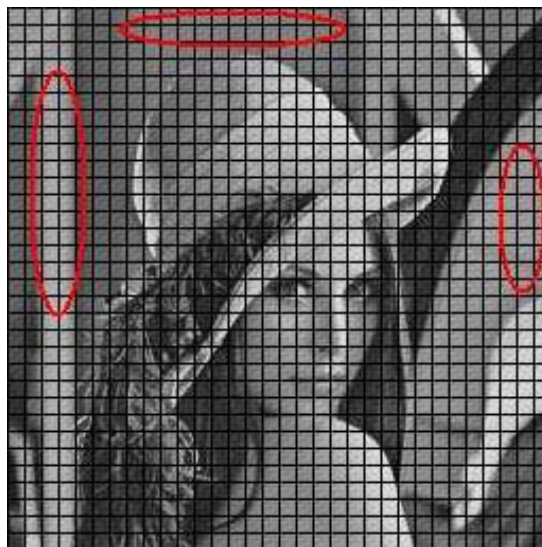


圖 1: 空間域上的區塊相似性 (圈選部份)

1 序論

靜態影像壓縮標準 JPEG [1],[6],[11] 是目前網路上及個人電腦中最常見到的壓縮格式, 其優點是有著較佳的編碼效能。JPEG 採用離散餘弦轉換 (Discrete cosine transform, DCT) [11]、量化 (Quantization)、霍夫曼編碼 (Huffman coding) 等技術。其中, 影像區塊經過 DCT 後, 可以將直流 (DC) 部分與交流 (AC) 部份分離出來, 區塊的最左上角係數為低頻, 也稱為直流係數, 其它則稱為 AC 係數。經過量化過程後, 將有許多係數變為 0, 此時利用 Z 字掃描 (Zig-zag scan) 來排序後使用可變長度編碼 (Variable length coding, VLC), 降低資料量。量化的作用是為了產生很多為 0 的係數, 因此, 量化表的定義就特別重要, 在 JPEG 標準 [2] 中的量化表, 其量化程度是由 Z 字掃描順序由小至大, 這樣可以確保在人眼敏感的低頻部份損失最少且達到良好的壓縮比。

霍夫曼編碼是屬於在 JPEG 中的熵編碼之一部分, 基本的原理是利用信號之統計特性, 將最常出現的符號指定最少位元編碼, 不常出現的符號則指定較多位元編碼。在 JPEG 標準中制定於檔案格式上面的霍夫曼編

碼表最多可以有四個, 分別是亮度直流、亮度交流、彩度直流、彩度交流部份。JPEG 壓縮法在區塊係數經量化後, 便以熵編碼減少資料量。但影像區塊間可能存在一些相似性, 如碎形編碼 [5] 的原理一樣, 影像內都有相似性的區域存在。從圖 1 中可以看出區塊間之相似性, 在頻率域上的 DCT 係數區塊也會有這樣的關係。

動態影像壓縮標準 MPEG-1/2 [3] 是目前最廣泛被應用來壓縮視訊, 前者常應用於 VCD 上, 後者則為 DVD 所採用。MPEG-1 中將一連串的畫面視為許多個訊框 (Frame), MPEG 中常以 GOP (Group of pictures) 當作一個場景畫面, GOP 包含了三個部份分別為訊框內畫面 (Intra-frame, I-frame), 預測畫面 (Predictive frame, P-frame), 雙向預測畫面 (Bi-predictive frame, B-frame)。其中 I-frame 採用的編碼方式與 JPEG 相似, 而 P-frame 則是參考上一張 I 或 P-frame 所得出的預測影像, B-frame 是參考 I 與 P-frame 所得出的預測影像, 這是因為在連續畫面中, 有許多相似不動的背景存在, 這些冗餘可以藉由預測編碼來去除。

本論文就區塊相似性並指定一個霍夫曼碼來指示編碼器是否採用索引值編碼, 並且建立於 JPEG/MPEG

的壓縮標準之上，在解碼時，索引值編碼的影像區塊，只需從前面複製對應的影像區塊即可，不但降低了整體的位元率，也可以略過逆 DCT 轉換等步驟，可提升解碼速度。另一方面，當使用錯誤的霍夫曼碼會造成熵解碼的錯誤蔓延，因此也提供影像內容保護的機制 [8]。

本論文之結構一開始是序論，接著為 JPEG/MPEG 壓縮標準之介紹，第三節提出改良 JPEG/MPEG 重建影像品質之演算法，第四節為實驗結果，第五節為本文的結論。

2 研究背景

2.1 JPEG壓縮

JPEG壓縮的過程為先判斷輸入的影像格式是否為全彩，若是則將 RGB 色彩轉換到 YCbCr 色彩空間，接著判斷輸入影像長寬是否為8或16 (依 CbCr 取樣率而定) 的倍數，若否則補齊後切割成為 8×8 的影像區塊，每個影像區塊像素值減去128之後並進行離散餘弦轉換，並除以一個量化表，此量化表會決定了壓縮比。壓縮的品質是藉由調整一個除以量化表的係數 Q 來控制， Q 值越大代表品質越好。接著利用 Z 字掃描分別排序後，以不同的霍夫曼表格做熵編碼 (直、交流部分不同，後面詳述)。這些 8×8 的區塊被稱作 DU (Data unit)，而組成要寫入的串流 (Stream) 格式則是以 MCU (Minimum coded unit) 作為最小單位，MCU 是由一個 DU 所組成，最後輸出 EOI (End of image) 就是 JPEG 標準壓縮格式。

2.2 熵編碼

在 JPEG 壓縮標準中，交流與直流係數是分開處理的，分別用差異脈衝編碼 (Differential pulse code modulation, DPCM) 與霍夫曼編碼，這個過程稱為熵編碼 (Entropy coding)。在直流係數中，步驟如下：

- 利用差異脈衝編碼，方法是將目前區塊直流係數減去上一區塊直流係數，利用相鄰區塊間常有的相關性，將其差異值編碼降低資料量。
- 利用直流霍夫曼編碼，其表格在 JPEG 壓縮標準中有兩個，一個是彩度表，一個是亮度表。
- 將編碼後的值儲存起來，待交流係數編碼完成後組成 DU，再由 DU 組成 MCU。

交流係數的處理方法與直流係數之編碼類似，不同之處是將差異脈衝編碼替換為可變長度編碼，直流霍夫

表 1: 可變長度編碼之長度

Value	Bit length
0	0
-1,1	1
-3,-2,2,3	2
-7,...-4,4,...,7	3
-15,...-8,8,...,15	4
-31,...-16,16,...,31	5
-63,...-32,32,...,63	6
-127,...-64,64,...,127	7
-255,...-128,128,...,255	8
-511,...-256,256,...,511	9
-1023,...-512,512,...,1023	10

曼編碼替換為交流霍夫曼編碼。由於經過量化後會產生很多個為0的係數，因此使用可變長度編碼可大大降低資料量。

2.2.1 可變長度編碼

可變長度編碼是將一個值 x 則劃分成

$\langle \text{RunLength}, \text{CodeLength} \rangle \langle x \rangle$ ，其中

RunLength 代表的是前面有幾個0，CodeLength 是這個值的長度 (參見表1)，而 x 代表要被編碼的值，這個值是2進位且負數需取1補數。還有兩個情況會發生：(1). 當0的個數超過15個。(2). 當0的個數為0且現在也為0。(0,0) 表示塊結束標記 (EOB)，(15,0) 表示有16個0 (Zero run length, ZRL)，當 RunLength 超過15時，用增加 ZRL 的個數來解決，所以最多有三個 ZRL ($3 \times 16 + 15 = 63$)。

2.2.2 霍夫曼編碼

霍夫曼編碼對機率高的符號分配較短的碼，反之亦然。JPEG標準制定的霍夫曼表為根據實驗所得出來的共通霍夫曼編碼表，對於大多數影像都有不錯的編碼效能。在壓縮影像過程所使用的霍夫曼編碼表，就會包含在檔頭 (Header) 中傳送給解碼端，在解碼時才能依照相同的霍夫曼編碼表解碼。表2列出了在直流係數部分的霍夫曼編碼表，分別對應亮度與彩度，左邊是輸入編碼值大小，其長度對應到的編碼值範圍如表1所示。右邊則是對應的亮度與彩度的編碼位元串流。更詳細的資訊可以在 JPEG 網站上查到[2]。

2.3 MPEG壓縮

進行 MPEG-1 壓縮前，先決定 I-、P-、B-frame 的數量，其編碼的順序為 I-、P-、B-frame，接著對 I-frame 進行

表 2: 直流係數霍夫曼編碼表

bit Length	Luminance	Chrominance
0	00	00
1	010	01
2	011	10
3	100	110
4	101	1110
5	110	11110
6	1110	111110
7	11110	1111110
8	111110	11111110
9	1111110	111111110
10	11111110	1111111110

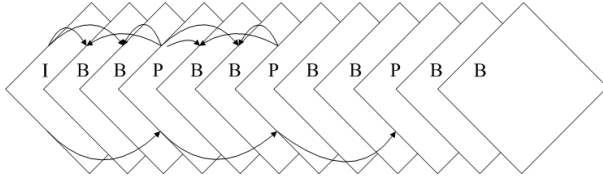


圖 2: I-, P-, B-frame 之排列順序與參考示意圖

切割區塊的動作, MPEG-1 中以巨方塊 (Macroblock) 為一個單位, 對每個巨方塊分成 4 個 8×8 的影像區塊進行 DCT 轉換並量化, 接著以類似 JPEG 架構之熵編碼降低資料量, 如此便完成 I-frame 之編碼。而 P-與 B-frame 格式進行移動估測 (Motion estimation) 後再進行編碼, 其中移動估測為 MPEG 系統中相當重要的一部分, 我們在接下來的小節詳細說明。

2.4 移動估測

編碼 P-與 B-frame 之前必須先對其進行移動估測, 目的是為了降低每個訊框之間的冗餘資料量。移動估測原始的方法為全域搜尋 (Full search), 但缺點是速度慢, 因此後來有許多改進的方法, 如區塊梯度搜尋 [4]、三步搜尋法等 [7]。移動估測的順序為 P-frame 參考前一張 I-或 P-frame, B-frame 則參考先前的 I-與 P-frame 來進行估測, 圖 2 為 I-, P-, B-frame 的排列順序與其參考的順序。

這裡以 P-frame 來說明移動估測的方式。首先將 P-frame 切割成不重疊的 16×16 巨方塊, 接著將每個巨方塊到參考訊框中搜尋最接近的影像巨方塊, 如果都無法找到理想匹配的巨方塊, 則 P-frame 中的此巨方塊以訊框內編碼 (Intra-coding) 處理, 反之, 則記錄參考訊框中的巨方塊索引值。B-frame 的移動估測類似於 P-frame, 只是參考訊框增加為兩張。

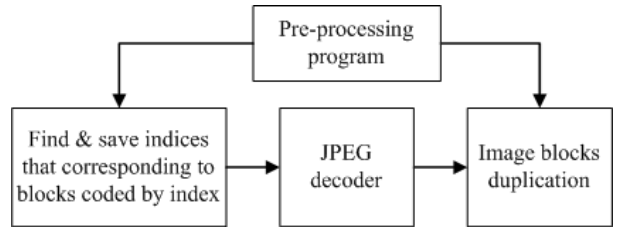


圖 3: 基於 JPEG/MPEG 之標準架構圖

3 改良 JPEG/MPEG 重建品質演算法

由於區塊 DCT 係數量化後大部分的係數值都成為 0, 因此在一張影像中, 會有許多相似的區塊, 若將這些冗餘的資料再次編碼, 將可再次降低編碼後的位元率, 換句話說相當於提升重建的品質。因此本論文利用修改的霍夫曼碼來進行是否為編碼相似區塊的指引, 並達到降低位元率的目的。另一方面, 由於霍夫曼編碼表的修改, 因此在解碼端, 若沒有使用正確的霍夫曼碼, 則解碼之影像將會嚴重失真, 所以, 本論文提出的演算法也提供了簡易的影像內容保護功能。

本論文提出的方法可以完全不需更改原有的 JPEG/MPEG 解碼器, 只需額外使用前處理程式便可以使用這兩項功能 (1) 提升重建影像品質. (2) 提供影像內容保護機制。這個架構如圖 3 所示, 分為三部分: (1) 在解碼前先判斷哪些是索引值編碼的影像區塊, 紀錄對應的索引值, 並將這些影像區塊的 DCT 係數設為零, 最後將霍夫曼表恢復成原始格式. (2) 輸入到一般 JPEG/MPEG 解碼器解碼, 此時解碼完成的影像, 在原本是用索引值編碼的影像區塊上會是一個黑色影像區塊. (3) 依照步驟 (1) 中紀錄的索引值, 到索引值對應的影像區塊複製所有的像素值即可。

本論文提出之演算法首先是在量化後經由 Z 字掃描排列後, 搜尋與目前編碼區塊係數之誤差 E 小於固定門檻值之區塊係數, 這個門檻值定義為誤差容忍度 T 。由於區塊經過 DCT 後係數重要性是由左上到右下遞減, 因此計算區塊間每點的權重也依此分配。 T 與 E 的計算在下一小節會提到。

本論文所提出的加密演算法步驟如下: (1). 尋找最相似的區塊並記錄其編號。(2). 選擇一個霍夫曼碼指示為索引值編碼。(3). 對有相似性的區塊給予該霍夫曼碼並只編碼其索引值。從文獻[10]中提到, 相似的區塊大多落在彼此附近, 基於這個概念, 在灰階影像中, 搜尋相似性的視窗大小為相鄰兩個區塊, 如圖 4 所示, 因此分配 1 bit 編碼索引值。這些架構並沒有修改 JPEG 壓縮

之標準，可視為系統的拓展選項。

3.1 區塊 DCT 係數誤差計算

這裡定義了 T 為兩個 DCT 係數的可容忍的差異程度， T 越大代表容忍誤差越高。經過 Z 字掃描排序之後的矩陣，內容越後面越不重要，因此為了更有效的計算 E ，本論文採取 $\log$ 以 10 為底的函數來表示權重 $\omega(i)$ 如下：

$$\omega(i) = \log_{10}(65 - i). \quad (1)$$

上式符合量化表的設計，權重的比例是隨著係數編號增加而遞減，如圖5所示。

區塊間的誤差 E 由下式表示：

$$E = \sum_{i=0}^{63} \omega(i) |D_n(i) - D_m(i)|, \quad (2)$$

其中 i 是指區塊中係數的順序，而 D_n 代表目前編碼的區塊， D_m 則是目標區塊， $n, m = \{1, 2, 3 \dots N_c\}$ ， N_c 則是區塊的總數，選擇 65 是為了避免在 $\log_{10}(1)$ 為 0 的情況。當 $E < T$ 的條件成立，代表這個 DCT 區塊係數將採用索引值編碼。

3.2 霍夫曼碼之設計

為了有效的提升重建影像品質，因此指示索引值編碼的霍夫曼碼便不能太長，本文採用的編碼表為直流霍夫曼碼。根據觀察，當霍夫曼碼為 00 的情況只出現在較暗的影像區塊上，因此在一般情形下，00 所用到的機率並不高，因此採用的霍夫曼碼為 00，並且其位元長度最低，效益最大。然而，隨著影像的內容不同，適當的選取霍夫曼碼可以得到較佳的效能。

接著，選取了霍夫曼碼之後，在 00 之後加入 1 bit 來判斷，此區塊 DCT 係數是否為索引值編碼，0 代表否，反之為是。因此直流霍夫曼編碼表可重新表示成如表 3 所示。

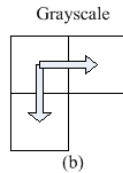


圖 4: 灰階影像之搜尋視窗

3.3 索引值的編碼

灰階影像的編碼索引值的方式是採用了類似可變長度編碼的格式，如下所示：

$$\langle \text{Huffman code} \rangle \langle \text{Flag} \rangle \langle \text{Index} \rangle, \quad (3)$$

其中 Huffman code 為指定的霍夫曼碼，而 Flag = 1 代表使用索引值編碼，固定為 1 bit，Index 則為 1 bit，0 表示右邊相鄰區塊，1 則是左邊相鄰區塊。

3.4 提升彩色 JPEG 影像品質

彩色影像的搜尋方式與灰階影像相當接近，在彩色影像中，一個像素點為 24 bits，因此共有三個分量可以尋找，分別是 Y、Cb、Cr 三個分量。接著，分別對三個分量分別進行相似性搜尋，每個分量分別指定相同或不同之霍夫曼碼來指示索引值解碼。

3.5 提升 MPEG 壓縮重建品質

提升 MPEG 重建視訊品質的方式是相當類似於彩色影像方法。圖 6 為提升 MPEG 重建視訊品質的流程圖，一開始只在 I-frame 中，整張影像使用彩色影像的搜尋視窗來搜尋相似區塊，並進行索引值的編碼，而 P-與 B-frame 則在未預測到的影像區塊群才進行搜尋的動作。如此一來 P-或 B-frame 這些預測影像就不需要花費太多時間在尋找相似 DCT 區塊係數上。

4 實驗結果

圖 7 為測試之灰階與彩色影像 Lena，影像大小為 512×512 。誤差容忍度 T 的門檻值設定為 0，影像品質係數 $Q = 2, 3, \dots, 10$ ，這裡的 Q 並非量化系數，設計上是量化表除上 Q ，因此 Q 越大則品質越好。從圖 8 中可以很明顯的看到，當採用相似性搜尋的方法時，相同的 PSNR

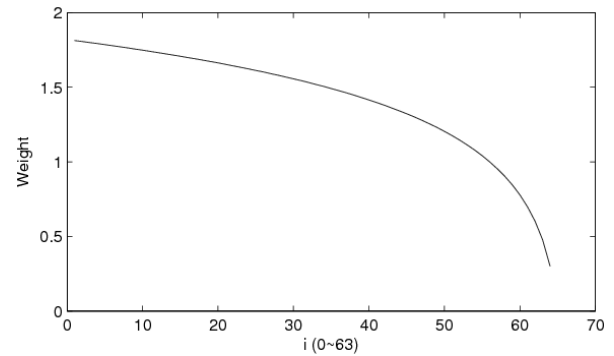


圖 5: 以 $\log_{10}$ 為基底之權重分配

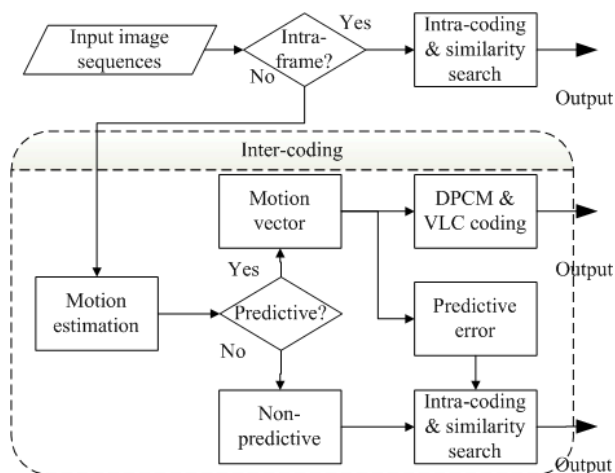


圖 6: 提升MPEG 重建視訊品質之流程圖

下可以提高壓縮率，換句話說，相同的壓縮率下本文所提出之方法可以明顯的改善重建影像品質。以彩色 Lena 影像測試也可以得出相同結果，如圖9所示，在設定的品質範圍內，使用相似搜尋方法可以對重建影像品質明顯的提升。

MPEG的測試方面採用圖10所示之影像序列，每個訊框大小為 256×256 之全彩影像，每秒15張之未壓縮視訊檔。從圖11中的測試結果可明顯的看出，本文所提出的方式可有效改進 MPEG-1解碼的重建視訊品質。

在解碼端時，使用索引值解碼之部分，只需要從已經解碼完成的區塊中複製過來，省略了逆向 DCT 等過程，可以提升解碼速度，如表4所示，使用本方法在 JPEG 與 MPEG 上都可以增加解碼速度。

4.1 影像內容保護

另一方面，當非法使用者欲解碼影像/視訊時，如果不知道正確的霍夫曼碼，則解碼後將無法辨識內容，但並非

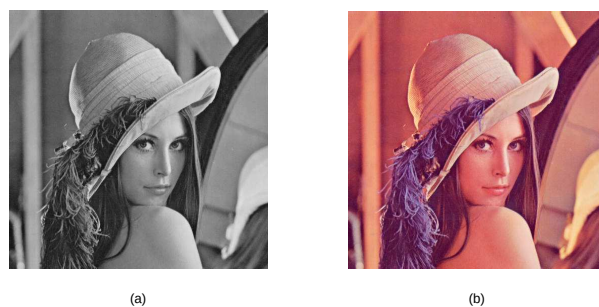


圖 7: 測試影像 (a) 灰階 Lena (b) 彩色 Lena

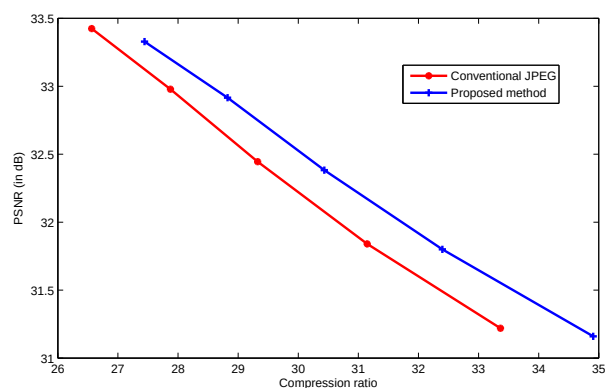


圖 8: 灰階Lena 之 PSNR 比較

不能解碼。因此，當使用者要使用加密功能時，僅需要指定特別建立或隨機選擇的霍夫曼碼，自己保留此項資訊即為密鑰。接著利用可信任的密鑰分配中心 (Key distribution center, KDC) 來傳遞密鑰給對方，其中 KDC 又有相當多種模式，這些都可以在許多密碼學之文獻中找到 [9]。圖12 (a) ~ (c) 分別為灰階 Lena, 彩色 Lena 與影像序列無正確霍夫曼碼解碼之結果，雖可正確解碼，但解碼影像卻已經無法辨識。因此除了重建影像的編碼效能改良，亦提供影像/視訊內容保護之功能。

表 3: 修正的直流係數霍夫曼編碼表

bit Length	Luminance	Chrominance
0	00(0/1)	00(0/1)
1	010	01
2	011	10
3	100	110
4	101	1110
5	110	111110
6	1110	11111110
7	11110	1111111110
8	111110	111111111110
9	1111110	11111111111110
10	11111110	1111111111111110

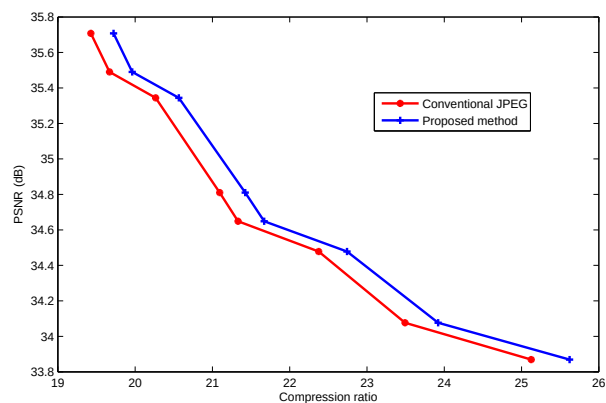


圖 9: 彩色Lena 之 PSNR 比較

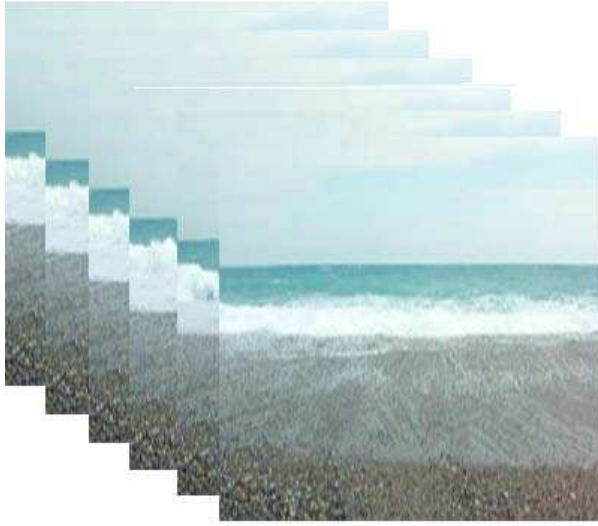


圖 10: 測試之影像序列

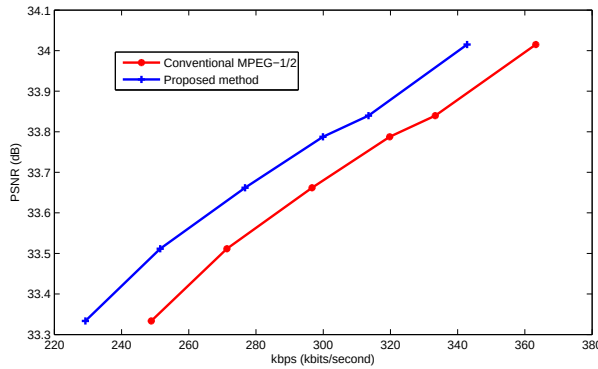


圖 11: 影像序列之PSNR 比較

5 結論

本論文所提出了一種基於 JPEG/MPEG 改善重建影像品質與提供保護影像內容的演算法。主要的兩點貢獻為 (1) 提升 JPEG/MPEG 的重建影像品質與加快其解碼的速度。 (2) 提供簡易的保護影像內容機制。更重要的是，本文所提之方法不需修改原有的 JPEG 解碼器，只需將影像經過索引值判斷的前處理，並在最後將影像區塊複製回來即可，因此，本論文提出之方法並沒有去修改 JPEG/MPEG 壓縮的標準，較具實用價值。

未來可將此觀念引用到 JPEG2000或是 MPEG-4 等其它進階壓縮技術上，來達到提升品質與強化保密的效果。

6 致謝

本論文為國科會研究計畫之補助，計畫編號為 NSC-95-2221-E-224-070-MY2，僅此致謝。

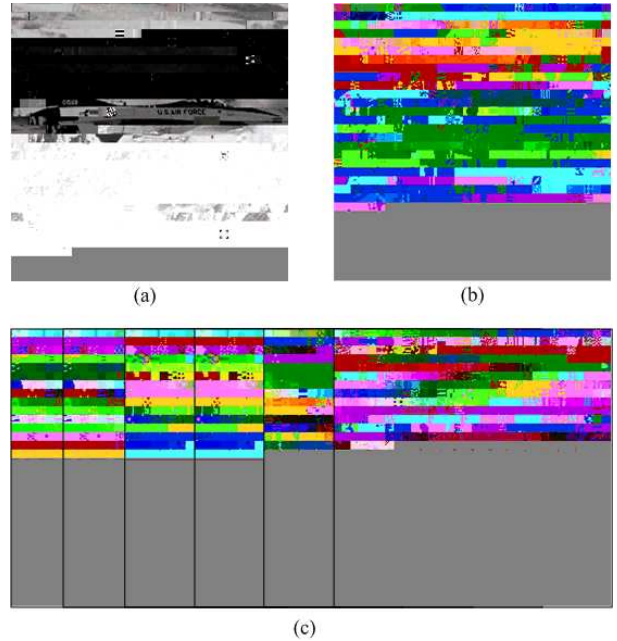


圖 12: 無正確霍夫曼碼之重建影像

表 4: JPEG與本論文方法之解碼時間比較

Image \ Time	JPEG	Proposed
Grayscale Lena	100 %	74.75 %
Grayscale Baboon	100 %	95.16 %
Color Lena	100 %	85.93 %
Color Baboon	100 %	90.04 %
Video \ Time	MPEG-1	Proposed
Image sequences	100 %	86.51 %

References

- [1] G.k. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. 18–34, No. 1, February 1992.
- [2] <http://www.jpeg.org/>
- [3] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, Second Edition, San Francisco, pp. 530–560, 2000.
- [4] L.L. Kuo, E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 6, pp. 419–422, August 1996.
- [5] M. Barnsley, "A better way to compress images," *BYTE archive*, vol. 13, pp. 215–223, January 1988.
- [6] Q. Sun, Q. Tian and S.F. Chang, "A robust and secure media signature scheme for JPEG images," *IEEE Transactions on Consumer Electronics*, vol. 4, pp. 296–299, April 2003.
- [7] R. Li, B. Zeng, M.L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Transactions On Circuits And Systems For Video Technology*, vol. 4, pp. 438–442, August 1994.
- [8] S. Sudharsanan, "Shared Key Encryption of JPEG Color Images," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 1204–1211, No. 4, November 2005.
- [9] W. Stallings, *Cryptography and network security principles and practices*, Person Education, Third Edition, Taipei, pp. 228–238, 2002.
- [10] X. Wu, D.J. Jackson, "Novel fractal image-encoding algorithm based on a full-binary-tree searchless iterated function system," *Optical Engineering*, vol. 44, October 2005.
- [11] Z.J. Liu, "Detection and correction of transmission errors in JPEG image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no.2, pp. 221–231, April 1998.