

第一章 序論

1.1 研究動機

數位照相機已經廣為大眾所接受，也漸漸取代傳統的底片照相機，下圖 1.1 為數位照相機的基本結構。

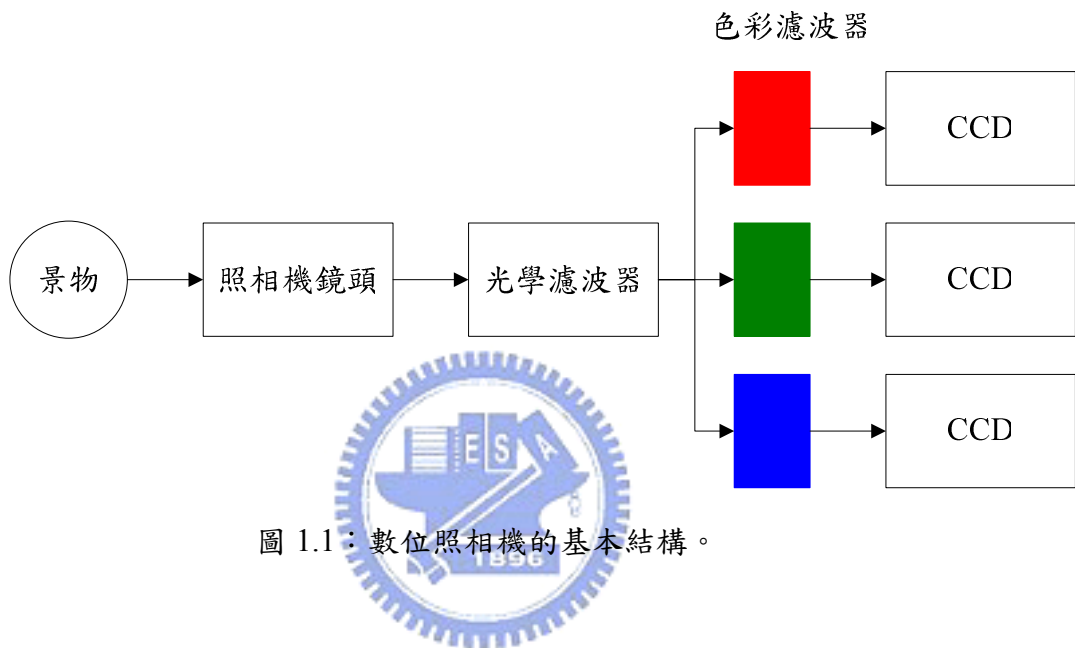


圖 1.1：數位照相機的基本結構。

景物的光反射經由照相機鏡頭和光學濾波器後被分開成三個部分，每一個部分只代表三原色中的一個刺激值(tristimulu)，使用這個基本結構的數位照相機對於一張全彩影像需要三個 CCD(Charged-Coupled Devices)，這樣是非常昂貴的，除了一般較專業的數位照相機才會使用如此的結構。現今一般我們所常用的數位照相機由於成本的問題通常只會使用一個 CCD 在搭配一個色彩濾波陣列 CFA(Color Filter Array)[1]如下圖 1.2：

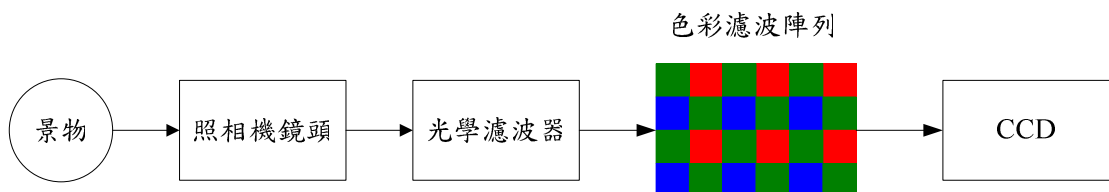


圖 1.2：單一影像感測器之數位照相機基本結構。

單一影像感測器之數位照相機在每一個像素只取三原色(R, G, B)中一個主要的顏色值，而色彩濾波陣列定義了每一個像素所要取樣的色彩，而擷取出的這張影像就叫作色彩濾波陣列影像或稱為馬賽克影像。一張全彩影像的再生過程即是將馬賽克影像中每一個像素值所缺少的其他兩個原色值估算出來。這個還原的過程一般通稱為色彩濾波陣列插分(CFA interpolation)。

有鑑於傳統單一影像感測器之數位照相機在壓縮的過程中，通常只將色彩濾波陣列插分後的全彩影像直接作壓縮在儲存到記憶體中。過程如下圖 1.3：

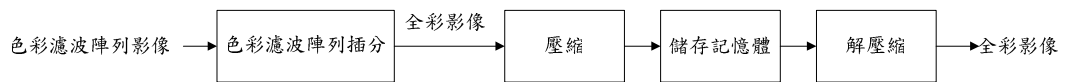


圖 1.3：傳統數位照相機壓縮與解壓縮的過程。

色彩濾波陣列影像提供了影像中邊緣的特性，若能將其特性使用在壓縮和解壓縮的過程，提出一種新的壓縮和解壓縮技術，如下圖 1.4，進而達到較好的壓縮率與影像的品質，此即為此篇論文的研究動機。再者由於在數位相機裡由馬賽克影像還原到全彩影像的過程是必要的，本論文只是將還原過程所需的每個像素邊緣的特性擷取出來，使用在壓縮的過程，也因此這部份的運算並不會因為要多計算光譜空間相關性演算法(SSC)而增加。

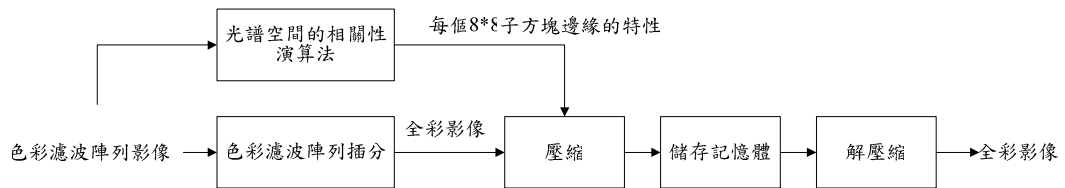


圖 1.4：新形數位照相機壓縮與解壓縮的過程。

最近幾年靜態影像壓縮的技術逐漸的進步中像 Joint Photographic Experts Group(JPEG)2000 再近幾年也被廣為探討，跟傳統的壓縮技術相比如 JPEG，它雖然可以用較小的 Bitrate(bit/pixel)達到相同的影像品質，但事實上現在的數位相

片並沒有支援這樣的格式，相對的 JPEG 卻是廣為被大家所使用。如現今的數位相機在要求的低成本與壓縮速度要快的情形下，所使用的壓縮技術就是 JPEG，畢竟 JPEG2000 在硬體的實作上是很耗成本的加上壓縮的速度也較傳統 JPEG 要慢的許多，在關鍵技術尚未突破之前 JPEG 還是廣被大家所使用的。也因此本論文將壓縮的重點鎖定在 JPEG 的技術上，希望藉由色彩濾波正列影像的資訊加入到傳統 JPEG 壓縮上進而達到較好的壓縮率與影像品質。

由於 JPEG 廣為大家所使用，因此有很多改良 JPEG 的方法被提出來討論。JPEG 壓縮將影像切割成每個 8×8 的子方塊，再分別對此方塊作編碼跟解碼的步驟，因此壓縮率越高，影像失真越多所造成的方塊效應就會越來越明顯。一部份的學者將 JPEG 壓縮的圖片經由他們個別所提出的一維或二維濾波器，藉此改善 JPEG 在高壓縮率下所造成的方塊效應，進而達到影像品質的增加[2-4]。另一部份的研究主要在針對編碼的部分，藉由加入統計的模組，降低一張圖片所需要的碼，藉此達到影像壓縮率的改善[5-7]。最後一部份的研究主要是加入圖片邊緣的資訊藉以改善傳統 JPEG 所造成的邊緣模糊化[8]。

1.2 問題描述

在第二章將介紹 JPEG 壓縮跟解壓縮的過程，可以知道照成影像失真的步驟在於量化的過程，傳統 JPEG 壓縮的量化表將影像中垂直的邊緣跟水平的邊緣做平均式的量化，這點可以從量化表中看出。試想如果影像中一個 8×8 的子方塊裡的垂直邊緣較多時，若還是對這一個 8×8 的子方塊的垂直跟水平方向作平均式的量化，這會造成垂直邊緣的部分產生較多的失真。從影像的圖片裡會發現垂直的邊緣會有模糊的現象，甚至於會使一些細微的邊緣不見了。

如何從色彩濾波陣列影像，利用光譜空間的相關性演算法得到一張影像中每一個 8×8 的子影像邊緣的特性，利用這些特性進而設計一套新的壓縮方法去解決傳統 JPEG 所造成的失真，這便是此篇論文的研究目標。

1.3 章節說明

本文的架構共分為 6 個章節，以下分別為各個章節內容的概要說明。

第二章介紹影像壓縮的基本原理以及 JPEG 壓縮與解壓縮的執行流程。第三章介紹如何使用光譜空間相關性(SSC)定義出影像中的水平邊緣集合，垂直邊緣集合和平緩邊緣集合。第四章介紹 Enhanced JPEG 是如何運用光譜空間相關性所定義的邊緣特性來對圖片作壓縮與解壓縮。第五章就 PSNR 與壓縮率比較傳統 JPEG 壓縮與 Enhanced JPEG 壓縮的方法。第六章為此論文的結論



第二章 JPEG 壓縮原理跟方法

本章主要的內容是根據[9-12]等書籍，作重點式的整理，其目的在於介紹傳統 JPEG(Joint Photographic Experts Group)壓縮的原理與方法。2.1 節中簡單的介紹 JPEG 的由來。2.2 節則介紹 JPEG 壓縮的流程。2.3 節介紹 JPEG 的解壓縮流程。2.4 節指出傳統 JPEG 在壓縮與解壓縮造成邊緣模糊化失真的缺點。

2.1 JPEG 概述

JPEG 是由全球各地的影像專業處理人員(主要來自 ISO 與 CCITT 兩個組織)，於 1986 年組合成立的機構，此機構致力於靜態影像壓縮標準的建立，JPEG 是在 1991 年公布的標準亦簡稱為 JPEG。

JPEG 影像壓縮法能提供良好的壓縮效果，因此為目前最廣為使用的影像壓縮法。JPEG 所採取的步驟是將一影像從空間域表示的形式，轉換到頻率域表示形式。一般而言，影像中高頻部分的係數值會比低頻部分的係數值要小許多。由於人類的眼睛，對於影像內高頻部分的變化較不敏感，所以高頻部分就被給予較大的量化來處理，使高頻係數的部分產生較多的 0，增加資料的抗餘性，因此可以大量的減低儲存該影像的記憶體空間。

JPEG 發展至今主要有以下四種模式：基本壓縮模式，漸進壓縮模式，失真壓縮模式，階層壓縮模式，其中以基本壓縮模式最常被使用。基本壓縮模式是一種失真性的壓縮，將原始影像資料壓縮再解壓縮會產生一部份的失真。影像雖然有小部分的失真但卻可以產生較大的壓縮率。

2.2 JPEG 壓縮流程

JPEG 主要壓縮步驟如下所示：

1. 色彩空間轉換：RGB 轉到 YCbCr。

2. Y 作位移，Cb 和 Cr 作取樣(down sampling)。
3. 將影像分割成許多個含有 8*8 個像素區塊。
4. 將這些區塊以離散餘弦轉換方法，從空間域影像轉到頻率域影像。
5. 再將這些二維的 DCT 係數，以 8*8 的量化矩陣做量化處理。通常高頻部分的 DCT 係數比低頻部分的 DCT 係數值較小，且人類的視力對高頻部分的變化有較低的敏感度。因此在量化時，可以對高頻部分的係數作較大的量化處理。如此使得在高頻部分的二維 DCT 係數，可以產生許多的零，以提高資料的抗餘性，同時增加資料的可壓縮性。這過程的處理會照成影像資料的失真。
6. 量化後的二維 DCT 係數，在經由 Z 字型係數排列 (zig-zag to sort)，將其轉換成一維陣列的資料。
7. 在經預先定義好的霍夫曼編碼表，來對這個一維陣列資料作霍夫曼編碼。因為 Z 字型係數排列 (zig-zag to sort) 的結果，使得高頻部分的零值緊靠在一起，在進行霍夫曼編碼時便可以得到進一步的壓縮。
8. 最後步驟產生 JPEG 檔案。

JPEG 壓縮流程圖如下圖 2.1 所示：

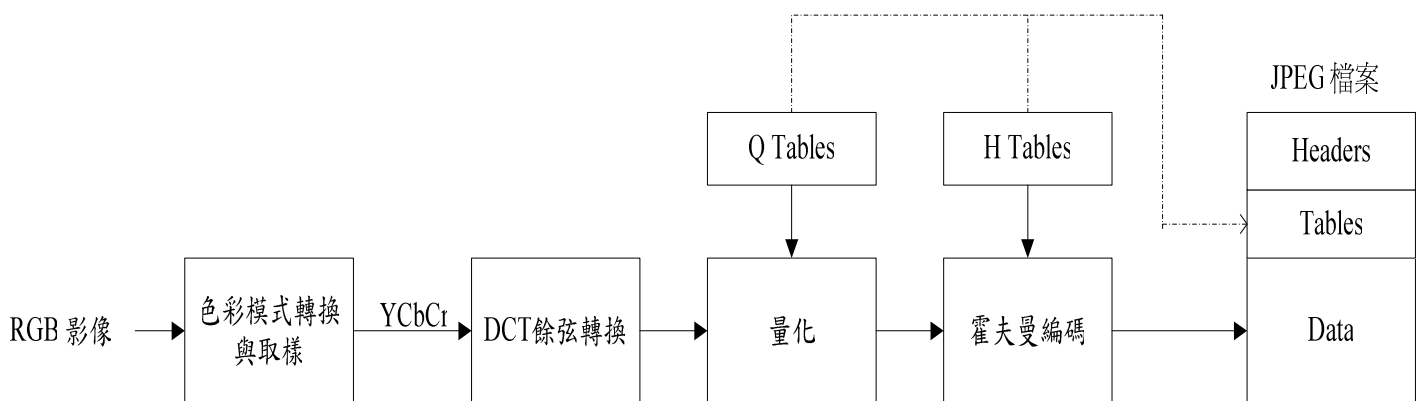


圖 2.1 JPEG 壓縮流程圖。

2.2.1 色彩模式將 RGB 色彩空間轉換到 YCbCr 色彩空間

為了配合目前的電腦輸出設備，所以原始影像的圖形檔案，會以 RGB 三個分量來做儲存，也就是說影像中每一個像素值(pixel)均以 Red, Green, Blue 三個顏色的分量來表示之，但是由於 RGB 三個分量的關連性很高，相同的資料常常會重複出現，造成了儲存空間的浪費。根據色彩學原理人類對色彩的感覺不像對亮度來的敏銳，所以必須要先將 RGB 色彩空間轉換到色彩與亮度分離的 YCbCr 色彩空間。如此一來便可以在 YCbCr 色彩空間中捨棄較多的色彩資料(Cb, Cr 分量)，而保留較完整的亮度部分(Y 分量)。RGB 色彩空間轉換到 YCbCr 色彩空間的轉換公式如下。

$$Y = 16 + 65.481 R + 128.553 G + 24.966 B$$

$$Cb = 128 - 37.797 R - 74.203 G + 112.000 B$$

$$Cr = 128 + 112.000 R - 93.786 G - 18.214 B$$

2.2.2 取樣(down sampling)與位移

現行 JPEG 影像壓縮標準的取樣主要有 YCbCr(4:2:2)跟 YCbCr(4:2:0)兩種。先以一塊 16*16 大小的影像區塊作說明，一塊 16*16 大小的影像區塊其實是由三個 16*16 的區塊所構成的，這三個區塊分別代表 RGB 三個分量，在經由上述色彩空間轉換後會得到三個區塊分別表示 YCbCr。這時保留完整的 Y 分量(即是四個 8*8 區塊的 Y 分量)，這時若 CbCr 分量在各別所屬的 16*16 區塊中以平均的方式各取得兩個 8*8 區塊的 CbCr 分量，此取樣即是 YCbCr(4:2:2)的取樣方法。若是 CbCr 分量在各別所屬的 16*16 區塊中以平均的方式各取得一個 8*8 區塊的 CbCr 分量，此取樣即是 YCbCr(4:2:0)的取樣方法。

一般而言 RGB 各分量大小的值在 0~255 之間在經過色彩空間轉換到 YCbCr 後 Y 分量的大小值依舊落於 0~255 間，而 CbCr 大小值的範圍卻在 -128~127 中間，為了使 YCbCr 各分量大小的值落在同一個範圍中將 Y 分量的大小做位移，

即是將 Y 分量中的每一個像素(pixel)值都減去 128。

2.2.3 DCT 離散餘弦轉換

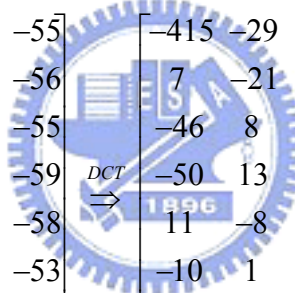
影像首先被分成大小為 8*8 的像素區塊，這是從左而右，從上到下處理的。然後對每一個區塊作二維的離散餘弦轉換。順向的離散餘弦轉換公式如下：

$$F(u, v) = \frac{1}{4} C(u) C(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$C(u) C(v) = \frac{1}{\sqrt{2}} \quad \text{for } u, v = 0$$

$$C(u) C(v) = 0 \quad \text{for others}$$

經過 DCT 轉換後的 8*8 矩陣稱為 DCT 矩陣，以一個 8*8 矩陣經過 DCT 轉換的結果來做說明：

-76	-73	-67	-62	-58	-67	-64	-55		-415	-29	-62	25	-55	-20	-1	3
-65	-69	-62	-38	-19	-43	-43	-56		7	-21	-62	9	11	-7	-6	6
-66	-69	-60	-15	16	-24	-24	-55		-46	8	77	-25	-30	10	7	-5
-65	-70	-57	-6	26	-22	-22	-59		-50	13	35	-15	-9	6	0	3
-61	-67	-60	-24	-2	-40	-40	-58		11	-8	-13	-2	-1	1	-4	1
-49	-63	-68	-58	-51	-65	-65	-53		-10	1	3	-3	-1	0	2	-1
-43	-57	-64	-69	-73	-67	-67	-45		-4	-1	2	-1	-2	-3	1	-2
-41	-49	-59	-60	-63	-52	-52	-34		-1	-1	-1	-2	-1	-1	0	-1

經由轉換後的矩陣，在最左上角的係數值通常是整個矩陣係數中最大的一個，這個值稱為 DCT 矩陣中的直流成分(DC component)，而其他的 63 個係數則稱為 DCT 矩陣中的交流成分(AC component)。在 DCT 矩陣中 DC 值為轉換後頻率為零的部份，越靠近 DC 值的幾個 AC 值係數表示頻率較低的部份，而越往右下角方向的 AC 值，其頻率越高。一般真實景物類的影像其色調跟亮度的變化較為緩和且大部分的訊號資訊都在低頻的部份，JPEG 壓縮的原理即是利用這個觀念大幅的刪減高頻的資訊使得還原後的影像與原始影像間的差異在人視覺可以接受的範圍。

2.2.4 量化

量化矩陣也是一個 8*8 的矩陣，它的 64 個值皆是預設的整數，對應 DCT 矩陣的 64 個能量係數，JPEG 標準規格中有一組標準的量化矩陣提供給使用者參考，但不見得適用於全部的圖形，使用者可以視情況自訂其量化矩陣。量化的方法是將 DCT 矩陣中的 64 個係數個別除以量化矩陣中相對應的整數，在取其最接近商數的整數，也因為如此的作法使得在做反量化時會照成資料的誤差。量化公式如下：

$$\text{量化後矩陣} = \text{round} \left[\frac{\text{DCT 矩陣}}{\text{量化矩陣}} \right]$$

JPEG 標準規格中提到的量化矩陣如圖 2.2：

亮度(luminance)量化矩陣	色彩(chrominance)量化矩陣																																																																																																																																
<table style="width: 100%; border-collapse: collapse;"> <tr><td>16</td><td>11</td><td>10</td><td>16</td><td>24</td><td>40</td><td>51</td><td>61</td></tr> <tr><td>12</td><td>12</td><td>14</td><td>19</td><td>26</td><td>58</td><td>60</td><td>55</td></tr> <tr><td>14</td><td>13</td><td>16</td><td>24</td><td>40</td><td>57</td><td>69</td><td>56</td></tr> <tr><td>14</td><td>17</td><td>22</td><td>29</td><td>51</td><td>87</td><td>80</td><td>62</td></tr> <tr><td>18</td><td>22</td><td>37</td><td>56</td><td>68</td><td>109</td><td>103</td><td>77</td></tr> <tr><td>24</td><td>35</td><td>55</td><td>64</td><td>81</td><td>104</td><td>113</td><td>92</td></tr> <tr><td>49</td><td>64</td><td>78</td><td>87</td><td>103</td><td>121</td><td>120</td><td>101</td></tr> <tr><td>72</td><td>92</td><td>95</td><td>98</td><td>112</td><td>100</td><td>103</td><td>99</td></tr> </table>	16	11	10	16	24	40	51	61	12	12	14	19	26	58	60	55	14	13	16	24	40	57	69	56	14	17	22	29	51	87	80	62	18	22	37	56	68	109	103	77	24	35	55	64	81	104	113	92	49	64	78	87	103	121	120	101	72	92	95	98	112	100	103	99	<table style="width: 100%; border-collapse: collapse;"> <tr><td>17</td><td>18</td><td>24</td><td>47</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>18</td><td>21</td><td>26</td><td>66</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>24</td><td>26</td><td>56</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>47</td><td>66</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> <tr><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td></tr> </table>	17	18	24	47	99	99	99	99	18	21	26	66	99	99	99	99	24	26	56	99	99	99	99	99	47	66	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99	99
16	11	10	16	24	40	51	61																																																																																																																										
12	12	14	19	26	58	60	55																																																																																																																										
14	13	16	24	40	57	69	56																																																																																																																										
14	17	22	29	51	87	80	62																																																																																																																										
18	22	37	56	68	109	103	77																																																																																																																										
24	35	55	64	81	104	113	92																																																																																																																										
49	64	78	87	103	121	120	101																																																																																																																										
72	92	95	98	112	100	103	99																																																																																																																										
17	18	24	47	99	99	99	99																																																																																																																										
18	21	26	66	99	99	99	99																																																																																																																										
24	26	56	99	99	99	99	99																																																																																																																										
47	66	99	99	99	99	99	99																																																																																																																										
99	99	99	99	99	99	99	99																																																																																																																										
99	99	99	99	99	99	99	99																																																																																																																										
99	99	99	99	99	99	99	99																																																																																																																										
99	99	99	99	99	99	99	99																																																																																																																										

圖 2.2 JPEG 標準量化矩陣。

由標準量化矩陣中可以看出，量化矩陣中整數的大小是取決於 DCT 矩陣中相對應係數在影像中的重要性。若該係數代表的是影像中的細節(DCT 矩陣中靠近右下角高頻的部份)，因為能量較小，被乎略後對影像品質不會影響很大，因此可以給予較大的除數使該係數在量化後所得的商數為零，以方便在下一部編碼的過程中照成較大的資料壓縮。

2.2.5 Z 字型係數排序 (zig-zag to sort)

當每一個 8*8 的 DCT 矩陣係數被量化後，將個別的 DCT 矩陣依照下圖 2.3 的 Z 字型圖樣重新排列

0	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 2.3 JPEG 之 Z 字型係數排列序列。

Z 字型係數排序將原本 8*8 的二維 DCT 矩陣轉換成 1*64 的一維係數序列。

以下圖 2.4: 8*8 DCT 矩陣為例：

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	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	0	0	0
0	0	0	0	0	0	0	0

圖 2.4 : 8*8 DCT 矩陣。

圖 2.4 : 8*8 DCT 矩陣根據圖 2.3 的 Z 字型重新排序後所得的一維係數序列為

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

其中符號 EOB 表示一個方塊結束(end-of-block)條件。一個特別的霍夫曼字碼用來表示在重排序列其餘的係數均為零。

2.2.6 Huffman 編碼

Huffman 編碼的演算法流程如下：

1. 將上述重排序列的 DC 值與 AC 值分開處理。
2. 相鄰兩個矩陣的 DC 值採用 DPCM(Differential Pulse Code Modulation)編碼。
3. 其他的 63 個 AC 值，將非零係數前面零值係數的個數及非零係數的大小類別去做查表編碼。



2.2.6.1 DC 值的編碼

對重排係數序列的預設 JPEG 碼的建構，是從對目前的 DC 係數與前一個已編碼過的子影像的 DC 係數(稱為預測值)間的差值計算開始。而影像中第一個 8*8 的方塊影像中的預測值定義為零。再舉列一個簡單的例子如下來幫助瞭解預測值的作法：

相鄰 DC 值：	2	6	7	8	5	2	0	4
-) 預測值：	<u>0</u>	<u>2</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>5</u>	<u>2</u>	<u>0</u>
DPCM :	2	4	1	1	-3	-3	-2	4

得到 DPCM 後只要將 DPCM 對事先定義好的 JPEG 霍夫曼編碼表做查表的動作就可以得到個別的 JPEG 霍夫曼碼。查表編碼的步驟如下：

1. 將 DPCM 對照表 2.1 看此值若於 DC 差值的哪一個類別。
2. 亮度分量 Y 按照表 2.2(色彩分量 CbCr 對照表 2.3)找出所在類別的基本碼 (base_code)。
3. 若 DPCM 值 ≥ 0 則其數值碼(value_code)為 DPCM 值的二進位碼。
4. 若 DPCM 值 ≤ 0 則其數值碼(value_code)為 DPCM 絕對值的二進位碼再取其一的補數。
5. 將基本碼與其數值碼結合則為 DPCM 的 JPEG 霍夫曼碼[base_code, value_code]

表 2.1：JPEG 係數編碼類別

範圍	DC 差值類別	AC 類別
0	0	N/A
-1, 1	1	1
-3,-2, 2,3	2	2
-7, ..., -4, 4, ..., 7	3	3
-15, ..., -8, 8, ..., 15	4	4
-31, ..., -16, 16, ..., 31	5	5
-63, ..., -32, 32, ..., 63	6	6
-127, ..., -64, 64, ..., 127	7	7
-255, ..., -128, 128, ..., 255	8	8
-511, ..., -256, 256, ..., 511	9	9
-1023, ..., -512, 512, ..., 1023	A	A
-2047, ..., -1024, 1024, ..., 2047	B	B

表 2.2：JPEG 預設的亮度 DC 碼

類別	基本碼	長度	類別	基本碼	長度
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

表 2.3：JPEG 預設的彩度 DC 碼

類別	基本碼	長度	類別	基本碼	長度
0	00	2	6	111110	12
1	01	3	7	1111110	14
2	10	4	8	11111110	16
3	110	6	9	111111110	18
4	1110	8	A	1111111110	20
5	11110	10	B	11111111110	22

以 $DPCM = 2$ 為例：對照表 2.1 可以知道他是屬於 DC 差值的第二個類別，在對照表 2.2 可以得知第二類別的基本碼為 [1 0 0]，DPCM 的值 $2 \geq 0$ 所以其數值碼為 [1 0]，結合基本碼跟數值碼可以得到 $DPCM = 2$ 的 JPEG 霍夫曼碼為 [1 0 0 1 0]。



2.2.6.2 AC 值編碼

AC 值編碼跟 DC 值編碼類似，主要的差別在於每一個預設 AC 值的基本碼取決於被編碼的非零係數前面的零值個數及其非零係數值的大小類別去做查表。方法步驟如下：

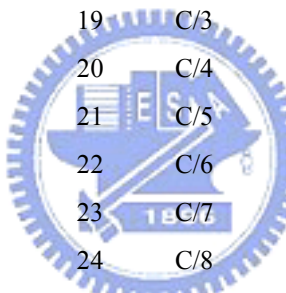
1. 將非零 AC 係數對照表 2.1 看此值若於 AC 類別的哪一個類別。
2. 計算個別非零 AC 係數前面連續零值係數的個數(稱之為同色個數)。
3. 亮度 Y 分量利用同色個數跟類別對非零的 AC 係數查表 2.4(色彩 CbCr 查表 2.5)找出個別的基本碼(base_code)。
4. 若 AC 值 > 0 則其數值碼(value_code)為 AC 值的二進位碼。
5. 若 AC 值 < 0 則其數值碼(value_code)為 AC 絕對值的二進位碼再取其一的補數。

6. 將基本碼與其數值碼結合則為 AC 的 JPEG 霍夫曼碼[base_code, value_code]
7. 在重排序列中若非零數值後面的其餘係數均為零值，則給其所有的零值一個特殊碼(EOB 碼)定義為[1 0 1 0]。此碼代表著每一個 8*8 方塊的重排一維序列的結束。

表 2.4：JPEG 預設亮度 AC 碼

同色/ 類別	基本碼	長度	同色/ 類別	基本碼	長度
0/0	1010(=EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	11111111000000	17
0/3	100	6	8/3	111111110110111	19
0/4	1011	8	8/4	111111110111000	20
0/5	11010	10	8/5	111111110111001	21
0/6	111000	12	8/6	111111110111010	22
0/7	1111000	14	8/7	111111110111011	23
0/8	111110110	18	8/8	111111110111100	24
0/9	111111110000010	25	8/9	111111110111101	25
0/A	111111110000011	26	8/A	111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	111111110111111	18
1/3	1111001	10	9/3	111111111000000	19
1/4	111110110	13	9/4	111111111000001	20
1/5	11111110110	16	9/5	111111111000010	21
1/6	111111110000100	22	9/6	111111111000011	22
1/7	111111110000101	23	9/7	111111111000100	23
1/8	111111110000110	24	9/8	111111111000101	24
1/9	111111110000111	25	9/9	111111111000110	25
1/A	111111110001000	26	9/A	111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	111111111001000	18
2/3	1111110111	13	A/3	111111111001001	19
2/4	11111111001001	20	A/4	111111111001010	20
2/5	11111111001010	21	A/5	111111111001011	21
2/6	11111111001011	22	A/6	111111111001100	22

2/7	11111111001100	23	A/7	111111111001101	23
2/8	11111111001101	24	A/8	111111111001110	24
2/9	11111111001110	25	A/9	111111111001111	25
2/A	11111111001111	26	A/A	111111111010000	26
3/1	111010	7	B/1	11111010	10
3/2	111110111	11	B/2	111111111010001	18
3/3	1111110111	14	B/3	111111111010010	19
3/4	111111110010000	20	B/4	111111111010011	20
3/5	111111110010001	21	B/5	111111111010100	21
3/6	111111110010010	22	B/6	111111111010101	22
3/7	111111110010011	23	B/7	111111111010110	23
3/8	111111110010100	24	B/8	111111111010111	24
3/9	111111110010101	25	B/9	111111111011000	25
3/A	111111110010110	26	B/A	111111111011001	26
4/1	111011	7	C/1	111111010	11
4/2	111111000	12	C/2	111111111011010	18
4/3	111111110010111	19	C/3	111111111011011	19
4/4	111111110011000	20	C/4	111111111011100	20
4/5	111111110011001	21	C/5	111111111011101	21
4/6	111111110011010	22	C/6	111111111011110	22
4/7	111111110011011	23	C/7	111111111011111	23
4/8	111111110011100	24	C/8	111111111100000	24
4/9	111111110011101	25	C/9	111111111100001	25
4/A	111111110011110	26	C/A	111111111100010	26
5/1	1111010	8	D/1	1111111010	12
5/2	111111001	12	D/2	111111111100011	18
5/3	111111110011111	19	D/3	111111111100100	19
5/4	111111110100000	20	D/4	111111111100101	20
5/5	111111110100001	21	D/5	111111111100110	21
5/6	111111110100010	22	D/6	111111111100111	22
5/7	111111110100011	23	D/7	111111111101000	23
5/8	111111110100100	24	D/8	111111111101001	24
5/9	111111110100101	25	D/9	111111111101010	25
5/A	111111110100110	26	D/A	111111111101011	26
6/1	1111011	8	E/1	11111110110	13
6/2	1111111000	13	E/2	111111111101100	18
6/3	111111110100111	19	E/3	111111111101101	19
6/4	111111110101000	20	E/4	111111111101110	20

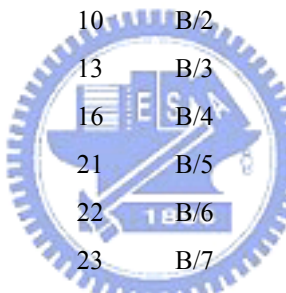


6/5	111111110101001	21	E/5	111111111101111	21
6/6	111111110101010	22	E/6	111111111110000	22
6/7	111111110101011	23	E/7	111111111110001	23
6/8	111111110101100	24	E/8	111111111110010	24
6/9	111111110101101	25	E/9	111111111110011	25
6/A	111111110101110	26	E/A	111111111110100	26
7/1	11111001	9	F/0	11111110111	12
7/2	1111111001	13	F/1	111111111110101	17
7/3	111111110101111	19	F/2	111111111110110	18
7/4	111111110110000	20	F/3	111111111110111	19
7/5	111111110110001	21	F/4	111111111111000	20
7/6	111111110110010	22	F/5	111111111111001	21
7/7	111111110110011	23	F/6	111111111111010	22
7/8	111111110110100	24	F/7	111111111111011	23
7/9	111111110110101	25	F/8	111111111111100	24
7/A	111111110110110	26	F/9	111111111111101	25
			F/A	111111111111110	26

表 2.5 : JPEG 預設彩度 AC 碼

同色/ 類別	基本碼	長度	同色/ 類別	基本碼	長度
0/0	00(=EOB)	2			
0/1	01	3	8/1	11111001	9
0/2	100	5	8/2	111111110110111	18
0/3	1010	7	8/3	111111110111000	19
0/4	11000	9	8/4	111111110111001	20
0/5	11001	10	8/5	111111110111010	21
0/6	111000	12	8/6	111111110111011	22
0/7	1111000	14	8/7	111111110111100	23
0/8	111110100	17	8/8	111111110111101	24
0/9	1111110110	19	8/9	111111110111110	25
0/A	111111110100	22	8/A	111111110111111	26
1/1	1011	5	9/1	111110111	10
1/2	111001	8	9/2	111111111000000	18
1/3	11110110	11	9/3	111111111000001	19
1/4	111110101	13	9/4	111111111000010	20
1/5	11111110110	16	9/5	111111111000011	21

1/6	11111110101	18	9/6	11111111000100	22
1/7	111111110001000	23	9/7	111111111000101	23
1/8	111111110001001	24	9/8	111111111000110	24
1/9	111111110001010	25	9/9	111111111000111	25
1/A	111111110001011	26	9/A	111111111001000	26
2/1	11010	6	A/1	11111000	10
2/2	11110111	10	A/2	111111111001001	18
2/3	111110111	13	A/3	111111111001010	19
2/4	11111110110	16	A/4	111111111001011	20
2/5	11111111000010	20	A/5	111111111001100	21
2/6	111111110001100	22	A/6	111111111001101	22
2/7	111111110001101	23	A/7	111111111001110	23
2/8	111111110001110	24	A/8	111111111001111	24
2/9	111111110001111	25	A/9	111111111010000	25
2/A	111111110010000	26	A/A	111111111010001	26
3/1	11011	6	B/1	11111001	10
3/2	11111000	10	B/2	111111111010010	18
3/3	111111000	13	B/3	111111111010011	19
3/4	11111110111	16	B/4	111111111010100	20
3/5	111111110010001	21	B/5	111111111010101	21
3/6	111111110010010	22	B/6	111111111010110	23
3/7	111111110010011	23	B/7	111111111010111	23
3/8	111111110010100	24	B/8	111111111011000	24
3/9	111111110010101	25	B/9	111111111011001	25
3/A	111111110010110	26	B/A	111111111011010	26
4/1	111010	7	C/1	11111010	10
4/2	11110110	11	C/2	111111111011011	18
4/3	111111110010111	19	C/3	111111111011100	19
4/4	111111110011000	20	C/4	111111111011101	20
4/5	111111110011001	21	C/5	111111111011110	21
4/6	111111110011010	22	C/6	111111111011111	22
4/7	111111110011011	23	C/7	111111111100000	23
4/8	111111110011100	24	C/8	111111111100001	24
4/9	111111110011101	25	C/9	111111111100010	25
4/A	111111110011110	26	C/A	111111111100011	26
5/1	111011	7	D/1	1111111001	12
5/2	111111001	12	D/2	111111111100100	18
5/3	111111110011111	19	D/3	111111111100101	19



5/4	111111110100000	20	D/4	111111111100110	20
5/5	111111110100001	21	D/5	111111111100111	21
5/6	111111110100010	22	D/6	111111111101000	22
5/7	111111110100011	23	D/7	111111111101001	23
5/8	111111110100100	24	D/8	111111111101010	24
5/9	111111110100101	25	D/9	111111111101011	25
5/A	111111110100110	26	D/A	111111111101100	26
6/1	1111001	8	E/1	1111111100000	15
6/2	1111110111	13	E/2	111111111101101	18
6/3	111111110100111	19	E/3	111111111101110	19
6/4	111111110101000	20	E/4	111111111101111	20
6/5	111111110101001	21	E/5	111111111110000	21
6/6	111111110101010	22	E/6	111111111110001	22
6/7	111111110101011	23	E/7	111111111110010	23
6/8	111111110101100	24	E/8	111111111110011	24
6/9	111111110101101	25	E/9	111111111110100	25
6/A	111111110101110	26	E/A	111111111110101	26
7/1	1111010	8	F/0	1111111010	11
7/2	1111111000	13	F/1	11111111000011	16
7/3	111111110101111	19	F/2	111111111110110	18
7/4	111111110110000	20	F/3	111111111110111	19
7/5	111111110110001	21	F/4	111111111111000	20
7/6	111111110110010	22	F/5	111111111111001	21
7/7	111111110110011	23	F/6	111111111111010	22
7/8	111111110110100	24	F/7	111111111111011	23
7/9	111111110110101	25	F/8	111111111111100	24
7/A	111111110110110	26	F/9	111111111111101	25
			F/A	111111111111110	26

以一維係數序列[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]為例：將上個一維數列中-1的數字作編碼：AC 值-1 對照表 2.1 可以知道他是屬於 AC 類別的第一個類別，再者-1 前面的連續零值個數為 2(同色個數=2)，利用同色個數跟類別對照表 2.4(色彩分量 CbCr 對照表 2.5)可以得到-1 的基本碼 [1 1 0 1 1]。AC 值-1 的數值碼為[0]，結合基本碼跟數值碼可以得到-1 的 JPEG 霍夫曼碼為[1 1 0 1 1 0]。

2.3 JPEG 解壓縮流程

JPEG 解壓縮的過程就是將其壓縮的過程反過來執行。解壓縮的流程如下圖

2.5 所示：

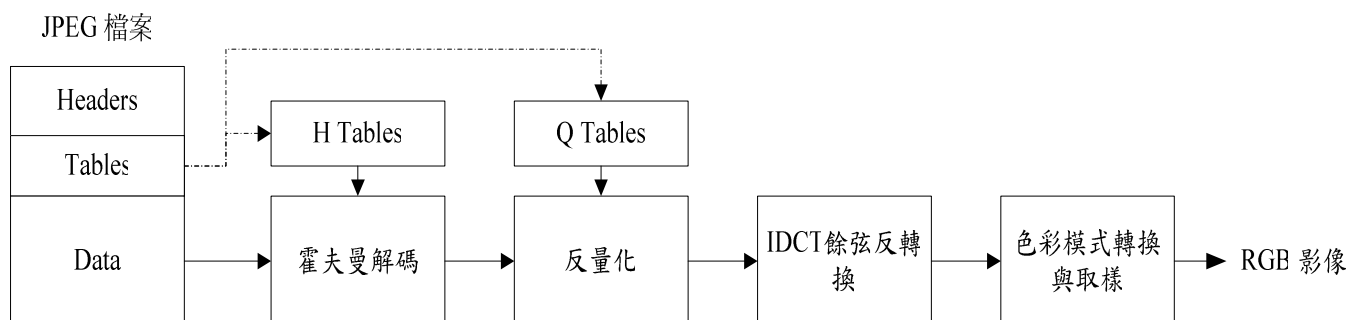


圖 2.5：JPEG 解壓縮流程

2.3.1 Huffman 解碼

因為一個霍夫曼編碼器所編出的 JPEG 霍夫曼碼是唯一的，所以解碼的步驟可以很容易用查表的方式來實現，解碼與編碼的過程類似 DC 值跟 AC 值也要分開處理。

2.3.1.1 DC 值解碼

1. 將壓縮後的資料跟表 2.2(CbCr 分量對照表 2.3)的基本碼做比對找出所在類別數值碼的長度。
2. 在壓縮資料裡再讀入數值碼長度個 bits。
3. 若讀入的第一個 bit 為 1，則這數值碼長度個 bit 所表示的值就是 DPCM。
4. 若讀入的第一個 bit 為 0，則需把這數值碼長度的 bit 反向後的值再加上負號才是 DPCM。
5. 求得所有的 DPCM 後，第一個 8*8 方塊的 DC 值即是所求得的第一個 DPCM 值，第二個 8*8 方塊的 DC 值即是所求得的第一個 DC 值加第二個 DPCM 值，

第三個 8*8 方塊的 DC 值即是所求得的第二個 DC 值加第三個 DPCM 值，以此類推可以得到下面的公式：

$$DC(i) = DPCM(i) + DC(i-1)$$

$$\text{其中 } DC(0) = 0$$

2.3.1.2 AC 值解碼

1. 將壓縮後的資料跟表 2.4(色彩 CbCr 查表 2.5)的基本碼做比對找出所在類別數值碼的長度與同色的個數。
2. 在壓縮資料裡再讀入數值碼長度個 bits。
3. 若讀入的第一個 bit 為 1，則這數值碼長度個 bit 所表示的值就是 AC 值。
4. 若讀入的第一個 bit 為 0，則需把這數值碼長度的 bit 反向後的值再加上負號才是 AC 值。
5. 若同色個數不為零時在填入 AC 值前必須先填入同色個數個零值。



2.3.2 Z 字型係數還原排列 (zig-zag to sort)

當每一個 1*64 的一維係數序列被解碼完成後，將個別的一維係數序列依照下圖 2.6 的 Z 字型圖樣重新排列還原成二維的 DCT 矩陣。

0	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 2.6 JPEG 之 Z 字型係數還原排列序列。

Z 字型係數還原排序將原本 1*64 的一維係數序列轉換成 8*8 的二維 DCT 矩陣。以下 1*64 的一維係數序列

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]為例：

一維係數序列依據圖 2.6 的 Z 字形重新排列還原後的 8*8DCT 矩陣如下圖 2.7

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	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	0	0	0
0	0	0	0	0	0	0	0

圖 2.7：8*8 DCT 還原矩陣。

2.3.3 量化反運算

在量化過程中，將量化矩陣的每一個元素分別除以量化矩陣中相對應的係數在取其最接近的商數，現在量化的反運算便是在解碼還原後的 DCT 矩陣中將每一個元素分別乘以量化矩陣中相對應的係數。以下圖 2.8 說明：圖 2.8 中量化過程使用圖 2.2 JPEG 標準量化矩陣中亮度(luminance)量化矩陣。由下圖 2.8 可以看出量化與反量化會照成一些資料的流失，這也是 JPEG 中影像失真的原因。

$$\begin{array}{ccc}
 \begin{bmatrix} -415 & -29 & -62 & 25 & -55 & -20 & -1 & 3 \\ 7 & -21 & -62 & 9 & 11 & -7 & -6 & 6 \\ -46 & 8 & 77 & -25 & -30 & 10 & 7 & -5 \\ -50 & 13 & 35 & -15 & -9 & 6 & 0 & 3 \\ 11 & -8 & -13 & -2 & -1 & 1 & -4 & 1 \\ -10 & 1 & 3 & -3 & -1 & 0 & 2 & -1 \\ -4 & -1 & 2 & -1 & -2 & -3 & 1 & -2 \\ -1 & -1 & -1 & -2 & -1 & -1 & 0 & -1 \end{bmatrix} & \xrightarrow{\text{量化}} & \begin{bmatrix} -26 & -3 & -6 & 2 & -2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 \\
 \begin{bmatrix} -416 & -33 & -60 & 32 & -48 & 0 & 0 & 0 \\ 12 & -24 & -56 & 0 & 0 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \xleftarrow{\text{反量化}} & \begin{bmatrix} -26 & -3 & -6 & 2 & -2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{array}$$

圖 2.8：量化與反量化的過程。

2.3.4 IDCT 離散餘弦反轉換

DCT 的反運算(IDCT)，其目的就是將空間頻率座標轉回圖面座標。其公式如下：

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$C(u)C(v) = \frac{1}{\sqrt{2}} \quad \text{for } u, v = 0$$

$$C(u)C(v) = 0 \quad \text{for others}$$

2.3.5 取樣還原(up sampling)與位移還原

現行 JPEG 影像壓縮標準的取樣主要有 YCbCr(4:2:2)跟 YCbCr(4:2:0)兩種。在做取樣還原時只要知道壓縮後 YCbCr 的影像比例就可以知道在做取樣時，我們是以 YCbCr(4:2:2)或是 YCbCr(4:2:0)哪一種取樣去做取樣，在還原的過程只要

用擴增的方式將 YCbCr 調成同樣大小的圖形即可。最後再將 Y 分量中的每一個像素(pixel)值都加上 128。

2.3.6 色彩模式轉換 YCbCr 色彩空間轉到 RGB 色彩空間

JPEG 影像還原的最後一個步驟就是將 YCbCr 色彩空間轉換回 RGB 色彩空間。轉換公式如下：

$$\begin{aligned} R &= -16 + 0.0046 Y + 0.0000 Cb + 0.0063 Cr \\ G &= -128 + 0.0046 Y - 0.0015 Cb - 0.0032 Cr \\ B &= -128 + 0.0046 Y + 0.0079 Cb + 0.0000 Cr \end{aligned}$$

2.4 JPEG 的缺點

就上述介紹 JPEG 壓縮跟解壓縮的過程可以知道，造成影像失真的步驟在於量化的過程，傳統 JPEG 壓縮的量化表將影像中垂直的邊緣跟水平的邊緣做平均式的量化，這點可以從量化表中看出。試想如果影像中一個 8*8 的子方塊裡的垂直邊緣較多時，若還是對這一個 8*8 的子方塊的垂直跟水平方向作平均式的量化，這會造成垂直邊緣的部分產生較多的失真。從影像的圖片裡會發現垂直的邊緣會有模糊的現象，甚至於會使一些細微的邊緣不見了。

第三章 光譜空間相關性(Spectral-Spatial Correlation)

本章主要的內容是根據[13]所提出的研究，作重點式的整理，其目的在於介紹本篇論文如何運用光譜空間相關性(Spectral-Spatial Correlation)來定義每一個 JPEG 壓縮的 8*8 子方塊其邊緣的特性。3.1 節中簡單的介紹現今單一影像感測器之數位照相機常用的 Bayer 色彩濾波陣列(Color Filter Array)[14]圖形。3.2 節則以 Bayer CFA 圖形簡單的介紹光譜相關性(Spectral Correlation)和空間相關性(Spatial Correlation)的定義。3.3 節中以光譜相關性和空間相關性為基礎，進一步定義光譜空間相關性(Spectral-Spatial Correlation ,SSC)的關係式。3.4 節則介紹如何使用 SSC 計算出每一個像素的邊緣特性，利用這個計算出來的結果便可以定義每一個 JPEG 壓縮的 8*8 子方塊邊緣的特性。

3.1 單一影像感測器之數位相機概述

一張全彩的影像在每一個像素(pixel)都含有三個數值，分別代表三原色(R, G, B)個別的分量。現今的數位照相機一般都採用單一影像感測器並搭配色彩濾波陣列在每一個像數只取三原色(R, G, B)中其中一個主要的顏色，這個目的在於節省成本，以及降低晶片的大小。

單一影像感測器之數位照相機在每一個像素只取三原色(R, G, B)中一個主要的顏色值，而色彩濾波陣列定義了每一個像素所要取樣的色彩。一張全彩影像的再生過程即是將每一個像素值所缺少的其他兩個原色值估算出來。這個還原的過程一般通稱為色彩濾波陣列插分(CFA interpolation)。

在所有現今發展的 CFA 圖形中，Bayer CFA[14]圖形是最常被使用的如下圖 3.1，由於綠色的元素對於一張彩色影像貢獻了較多亮度的成分，這也解釋了圖形中為何綠色分量較紅色與藍色分量多的原因(亮度的部分較容易影響一張影像視覺的品質)[14]。

R ₁₁	G ₁₂	R ₁₃	G ₁₄	R ₁₅
G ₂₁	B ₂₂	G ₂₃	B ₂₄	G ₂₅
R ₃₁	G ₃₂	R ₃₃	G ₃₄	R ₃₅
G ₄₁	B ₄₂	G ₄₃	B ₄₄	G ₄₅
R ₅₁	G ₅₂	R ₅₃	G ₅₄	R ₅₅

圖 3.1 : Bayer CFA pattern。

3.2 光譜相關性和空間相關性(Spectral and Spatial Correlation)

光譜相關性(Spectral Correlation)的概念是建立在下面的假設：不同色彩信號的差值在色彩平緩的區域中近似於常數[15]。更進一步明確的說，定義[R G B]是造成一張彩色影像的三種元素，光譜相關性的概念可以導致下面式(1)的結果。

$$R(x, y) = G(x, y) + A_{rg}(x, y) \quad \text{和} \quad B(x, y) = G(x, y) + A_{bg}(x, y) \quad (1)$$

其中的 $A_{rg}(x, y)$ 和 $A_{bg}(x, y)$ 在同一個邊界條件下是片斷的常數。

舉例說明：如上圖 3.1 中 R₃₃ 假設這個像數值是位於水平的邊緣可以得到下式(2)

$$\begin{aligned}
 A_{rg}(x-1, y) &= A_{rg}(x, y) = A_{rg}(x+1, y) \\
 A_{bg}(x-1, y) &= A_{bg}(x, y) = A_{bg}(x+1, y) \\
 \bar{R}_{32} - G_{32} &= R_{33} - \bar{G}_{33} = \bar{R}_{34} - G_{34} \\
 \bar{B}_{32} - G_{32} &= \bar{B}_{33} - \bar{G}_{33} = \bar{B}_{34} - G_{34}
 \end{aligned} \quad (2)$$

其中 $\bar{R}_{32}, \bar{G}_{33}, \bar{R}_{34}, \bar{B}_{32}, \bar{B}_{33}$ and \bar{B}_{34} 表示該像數中缺少的色彩值。

空間相關性(Spatial Correlation)說明沿著同一個方向邊緣，鄰近間的像素的差值在空間域上是一個常數。如上圖 3.1 中 R_{33} 假設這個像數值是位於水平的邊緣這個定義即說明有下關係式(3)的存在：

$$\begin{aligned}\bar{R}_{32} - R_{33} &= R_{33} - \bar{R}_{34} \equiv dR_h \\ G_{32} - \bar{G}_{33} &= \bar{G}_{33} - G_{34} \equiv dG_h \\ \bar{B}_{32} - \bar{B}_{33} &= \bar{B}_{33} - \bar{B}_{34} \equiv dB_h\end{aligned}\quad (3)$$

其中 $\bar{R}_{32}, \bar{G}_{33}, \bar{R}_{34}, \bar{B}_{32}, \bar{B}_{33}$ and \bar{B}_{34} 表示該像數中缺少的色彩值 dR_h , dG_h and dB_h 表示常數，而下標 h (horizontal)表示位於水平方向的邊緣。

3.3 光譜空間相關性(SSC)

光譜和空間相關性描述一張影像中不同色彩元素間的關係，然而在 Bayer CFA 圖形後取出的色彩值是很難直接計算其光譜和空間的相關性，原因在於每個像素只有一個主要的顏色，缺少了其他兩個主要的顏色。這個問題驅使我們在 Bayer CFA 圖形後取出的色彩值後尋找一個更有效率的標準去取代光譜和空間相關性。

仔細看 Bayer CFA 圖形可以發現一個明顯的特徵：對任一個像素而言，其周圍圍繞著不同的主要顏色分量。這導致我們研究在邊緣上鄰近的像素在不同色彩元素之間的關係。現在考慮在一個水平的邊緣下一個紅色的像素被兩個綠色的像素包圍，計算在水平邊緣條件下中間紅色像素跟右邊綠色像素間的差值如下式(4)：

$$R(x, y) - G(x+1, y) = [R(x, y) - \bar{G}(x, y)] + [\bar{G}(x, y) - G(x+1, y)] \quad (4)$$

其中 $\bar{G}(x, y)$ 定義為中間紅色像素所缺少的綠色分量部分。根據上式(1)·(3)式，(4)式可以得到如下的結果：

$$S_{rg}^{h(x, x+1)} \equiv R(x, y) - G(x+1, y) = A_{rg}(x, y) + dG_h \quad (5)$$

同理在一個水平的邊緣下一個藍色的像素被兩個綠色的像素包圍，計算在水平邊緣條件下中間藍色像素跟右邊綠色像素間的差值可得如下的結果：

$$S_{bg}^{h(x,x+1)} \equiv B(x,y) - G(x+1,y) = A_{bg}(x,y) + dG_h \quad (6)$$

同樣的理由與步驟，用在延著垂直邊緣的方向上可以如下的結果：

$$S_{rg}^{v(y,y+1)} \equiv R(x,y) - G(x,y+1) = A_{rg}(x,y) + dG_v$$

$$S_{bg}^{v(y,y+1)} \equiv B(x,y) - G(x,y+1) = A_{bg}(x,y) + dG_v \quad (7)$$

由(5)-(7)式可以得知不同色彩元素的鄰近像素的差值會等於其光譜和空間的相關性的和。這些關係便定義為光譜空間相關性(SSC)。SSC 有兩個重要的特性，第一，SSC 可以很容易且直接從 Bayer CFA 圖形後取出的色彩值直接計算。第二，SSC 延續了傳統光譜和空間相關性的特性，換句話說即是在同一個邊界條件下或沿著同一個邊緣 SSC 是片斷的常數。



3.4 異次投影(Heterogeneity-Projection)

本小節的目的在於從 SSC 的關係式介紹異次投影的方法，經由異次投影，可以把 Bayer CFA 圖形，直接轉換到水平跟垂直的異次係數矩陣。利用這兩個矩陣便可判斷每一個像數的垂直或水平邊緣的特性，進一步可以定義出 JPEG 壓縮中每一個 8*8 子方塊的邊緣特性。

假設上圖 3.1 中 R_{33} 位於水平的邊緣，以 R_{33} 周圍的像素分別定義一階水平 SSC 的有限導函數和二階水平 SSC 的有限導函數。由於 R_{33} 位於水平的邊緣所以由上 3.2 和 3.3 小節的介紹可以得到下式(8)：

$$S_{rg}^{h(1,2)} = A_{rg}(1,3) + dG_h = R_{31} - G_{32}$$

$$S_{gr}^{h(2,3)} = -A_{rg}(2,3) + dG_h = G_{32} - R_{33}$$

$$S_{rg}^{h(3,4)} = A_{rg}(3,3) + dG_h = R_{33} - G_{34}$$

$$S_{gr}^{h(4,5)} = -A_{rg}(4,3) + dG_h = G_{34} - R_{35} \quad (8)$$

定義一階水平 SSC 的有限導函數如下式(9)：

$$\begin{aligned} dS_{rg}^{h(1,4)} &\equiv S_{rg}^{h(1,2)} - S_{rg}^{h(3,4)} = A_{rg}(1,3) - A_{rg}(3,3) = R_{31} - G_{32} - R_{33} + G_{34} \\ dS_{gr}^{h(2,5)} &\equiv S_{gr}^{h(2,3)} - S_{gr}^{h(4,5)} = A_{rg}(4,3) - A_{rg}(2,3) = G_{32} - R_{33} - G_{34} + R_{35} \end{aligned}$$

$$\begin{bmatrix} dS_{rg}^{h(1,4)} & dS_{gr}^{h(2,5)} \end{bmatrix} = \begin{bmatrix} R_{31} & G_{32} & R_{33} & G_{34} & R_{35} \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ -1 & -1 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \quad (9)$$

由上式(9)可以知道沿著水平方向的邊緣，一階水平 SSC 的有限導函數 $dS_{rg}^{h(1,4)}$ 和 $dS_{gr}^{h(2,5)}$ 會趨近於零。

定義二階水平 SSC 的有限導函數如下式(10)：

$$d^2 S_{rg}^{h(1,5)} \equiv dS_{rg}^{h(1,4)} - dS_{gr}^{h(2,5)} = A_{rg}(1,3) + A_{gr}(2,3) - A_{rg}(3,3) - A_{gr}(4,3) \quad (10)$$

由上式也可以知道沿著水平方向的邊緣，二階水平 SSC 的有限導函數會更趨近於零。因此由上兩式可以得到如下的假設：如果像素位於一個方向上的邊緣則沿著這個邊緣的方向， n 階的 SSC 有限導函數會趨近於零。在此稱 n 階的 SSC 有限導函數為異次的值，因為它在一個方向性平緩的區域會導致一個很小的值。由上(9)式可知一個大小為 (1×5) 的向量可以找到兩個一階的水平 SSC 的有限導函數，若一個大小為 $(1 \times N)$ 的向量 $(RG_{1 \times N} = [R_1 \ G_2 \ R_3 \ \dots]_{1 \times N})$ 則可以找到一個 $(1 \times N-3)$ 的向量表示一階的 SSC 有限導函數。

現在令 $T_{N \times (N-3)}^1 = [1 \ -1 \ -1 \ 1] \otimes I(N-3)$ ，其中 \otimes 為二維的迴旋積運算子， $I(M)$ 表示 $M \times M$ 的單位矩陣，則一階的水平 SSC 的有限導函數可表示如下(11)式：

$$\begin{bmatrix} dS_{rg}^{h(1,4)} & dS_{gr}^{h(2,5)} & dS_{rg}^{h(3,6)} & \dots \end{bmatrix}_{1 \times (N-3)} = RG_{1 \times N} T_{N \times (N-3)}^1 \quad (11)$$

令 $T_{(N-3) \times 1}^2 = \prod_{i=1}^{N-4} [1 \quad -1]^T \otimes I(N-3-i)$ 則 n 階水平的 SSC 有限導函數為水平異質性的值 H_h 可表示如下(12)式

$$H_h = [dS_{rg}^{h(1,4)} \quad dS_{gr}^{h(2,5)} \quad dS_{rg}^{h(3,6)} \quad \dots]_{1 \times (N-3)} T_{(N-3) \times 1}^2 = RG_{1 \times N} T_{N \times (N-3)}^1 T_{(N-3) \times 1}^2 = RG_{1 \times N} P_{N \times 1} \quad (12)$$

其中 $P_{N \times 1} = T_{N \times (N-3)}^1 T_{(N-3) \times 1}^2$ 稱為異次向量，上式(12)可知 n 階水平的 SSC 有限導函數(水平異次的值 H_h)等於經由 Bayer CFA 圖形後取出的色彩值的列向量對異次向量作投影。經由上述的計算過程可得到每一個像素的水平異次值，如下式：

$$H_{h_map} = |Bayer \otimes P_{N \times 1}^T| \quad (13)$$

其中 Bayer 表示經由 Bayer CFA 圖形後取出的色彩值。

同理每一個像素垂直的異次值計算如下式(14)

$$H_{v_map} = |Bayer \otimes P_{N \times 1}| \quad (14)$$

得到 H_{h_map} 和 H_{v_map} 後，利用下式(15)定義出影像中的水平邊緣集合 Ω_h ，垂直邊緣集合 Ω_v 和平緩邊緣集合 Ω_s 為，

$$\begin{aligned} \Omega_h &\equiv \{(x, y) \mid H_{h_map}(x, y) < \alpha H_{v_map}(x, y)\} \\ \Omega_v &\equiv \{(x, y) \mid H_{v_map}(x, y) < \alpha H_{h_map}(x, y)\} \\ \Omega_s &\equiv \{(x, y) \mid (x, y) \notin \Omega_h, (x, y) \notin \Omega_v\} \end{aligned} \quad (15)$$

其中 α 表示介於 0~1 間的小數，本篇論文的 α 值設定為 0.55。

第四章 Enhanced JPEG 壓縮原理跟方法

本章節的目地在於如何在現有的 JPEG 基礎下，加入來自色彩濾波陣列影像所擷取出的邊緣特性，設計一套新的演算法藉此改善傳統 JPEG 造成邊緣模糊化的失真。4.1 節中概略的介紹 Enhanced JPEG 的方法與跟傳統 JPEG 的差異，4.2 節中細部的介紹整個 Enhanced JPEG 的壓縮流程，4.3 節則介紹 Enhanced JPEG 的解壓縮流程。

4.1 Enhanced JPEG 概述

Enhanced JPEG 壓縮方法，基本上是建構在傳統 JPEG 壓縮方法上的，唯一的差別在於將經過色彩空間轉換後的 Y(亮度分量)獨立出來做不同的運算。而兩者間對 Y 分量壓縮的主要差別是 Enhanced JPEG 壓縮方法會先將影像中每一個 Y 分量的 8*8 子方塊計算其水平跟垂直邊緣的比重，預先定義好一個門檻值，若水平邊緣的比重高於此門檻值，則此 8*8 的子方塊具有水平方向的相關性;反之若垂直邊緣的比重高於此門檻值，則此 8*8 的子方塊具有垂直方向的相關性;若水平邊緣跟垂直邊緣的比重都不高則此 8*8 的子方塊不具有水平或垂直的相關性。定義好每一個 8*8 子方塊的相關性後再對於所屬不同的相關性做不同的運算跟量化過程，使影像中水平跟垂直邊緣的細節更為明顯，進而完成影像品質的改善。

4.2 Enhanced JPEG 壓縮流程

Enhanced JPEG 主要壓縮步驟如下所示：

1. 色彩空間轉換：RGB 轉到 YCbCr。
2. 將亮度 Y 與色彩 CbCr 分開處理，其中 CbCr 部分依據傳統 JPEG 壓縮流程。
3. 將亮度 Y 影像分割成許多個含有 8*8 個像素區塊，定義每一個 8*8 的子方塊

水平跟垂直的相關性。

4. 8×8 的子方塊若屬於水平高相關性的對其做一維的水平 DCT 離散餘旋轉換。
5. 8×8 的子方塊若屬於垂直高相關性的對其做一維的垂直 DCT 離散餘旋轉換。
6. 8×8 的子方塊若屬於不具垂直或水平高相關性的對其做二維的 DCT 離散餘旋轉換。
7. 再將個別的 DCT 係數，對其所屬的 8×8 量化矩陣做量化處理。 8×8 的子方塊若屬於水平高相關性使用水平量化矩陣做量化處理， 8×8 的子方塊若屬於垂直高相關性使用垂直量化矩陣做量化處理， 8×8 的子方塊若屬於不具垂直或水平高相關性使用傳統 JPEG 量化矩陣做量化處理。
8. 量化後的 DCT 係數，在經由垂直重新排列，水平重新排列或 Z 字型係數重新排列，將其轉換成一維陣列的資料。 8×8 的子方塊若屬於水平高相關性使用水平重新排列，將其轉換成一維陣列的資料， 8×8 的子方塊若屬於垂直高相關性使用垂直重新排列，將其轉換成一維陣列的資料， 8×8 的子方塊若屬於不具垂直或水平高相關性使用，Z 字型係數重新排列，將其轉換成一維陣列的資料。
9. 在經預先定義好的霍夫曼編碼表，來對這個一維陣列資料作的霍夫曼編碼。

Enhanced JPEG 壓縮流程如下圖 4.1

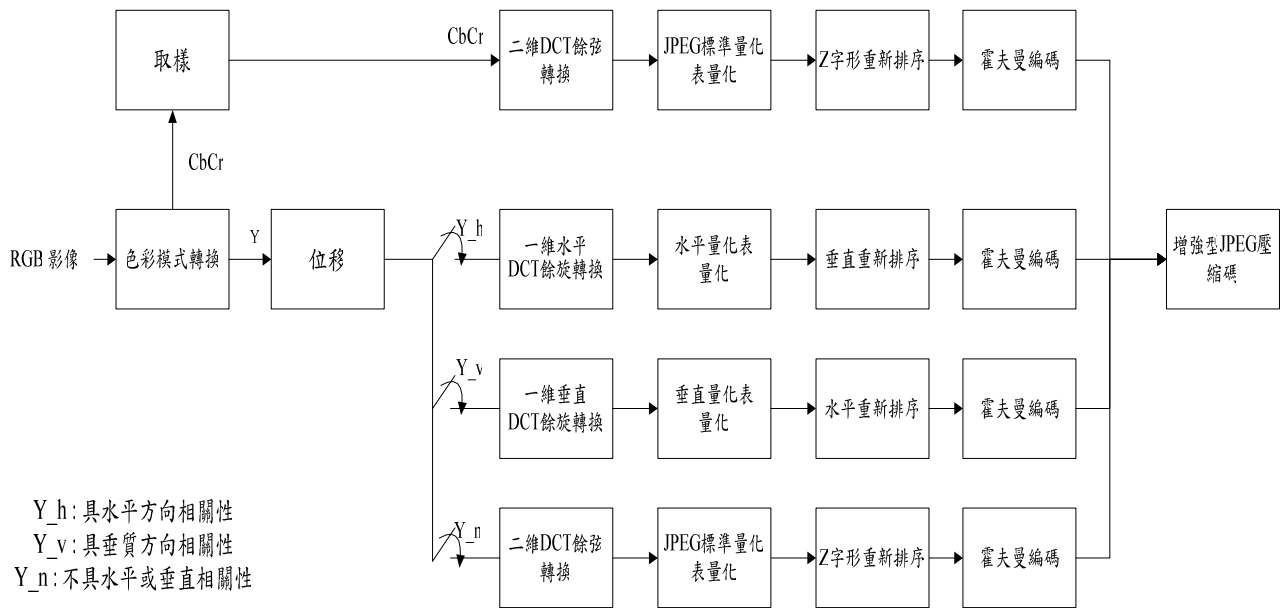


圖 4.1：Enhanced JPEG 壓縮流程圖。

4.2.1 一維水平跟垂直 DCT 離散餘旋轉換

影像首先被分成大小為 8*8 的像素區塊，這是從左而右，從上到下處理的。

一維垂直 DCT 離散餘旋轉換公式如下：

$$f(u, v) = \frac{1}{4} C(v) \sum_{y=0}^7 f(x, y) \cos \frac{(2y+1)v\pi}{16}$$

$$C(v) = \frac{1}{\sqrt{2}} \quad \text{for } v = 0$$

$$C(v) = 1 \quad \text{for } v \neq 0$$

8*8 的一維垂直 DCT 轉換，將 8*8 矩陣分成八個行向量(8×1)，第一個行向量 $x = u = 1$ ，第二個行向量 $x = u = 2$ 以此類推分別引用上述的公式即可得到一個 8*8 矩陣的一維垂直 DCT 轉換矩陣。

一維水平 DCT 離散餘旋轉換公式如下：

$$f(u, v) = \frac{1}{4} C(u) \sum_{x=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16}$$

$$C(u) = \frac{1}{\sqrt{2}} \quad \text{for } u = 0$$

$$C(u) = 1 \quad \text{for } u \neq 0$$

同理 8*8 的一維水平 DCT 轉換，將 8*8 矩陣分成八個列向量(1*8)，第一個列向量 $y = v = 1$ ，第二個列向量 $y = v = 2$ 以此類推分別引用上述的公式即可得到一個 8*8 矩陣的一維水平 DCT 轉換矩陣。

以一個具有垂直相關性 8*8 子方塊如下圖 4.2 為例：對此矩陣做一維垂直離散餘旋轉換結果如下圖 4.3，做一維的水平離散餘旋轉換如下圖 4.4，各轉換後的矩陣係數小數點後四捨五入。

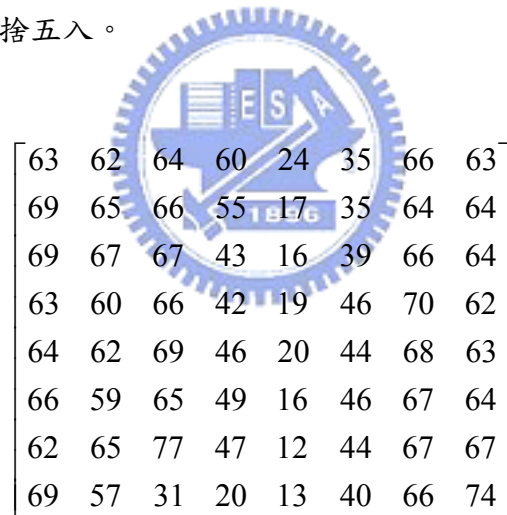


圖 4.2：垂直相關性 8*8 子方塊矩陣。

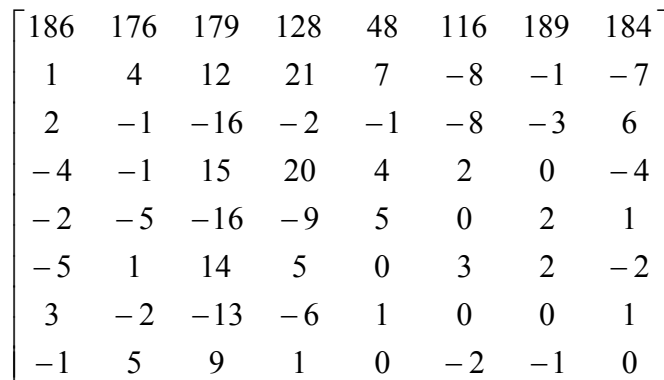


圖 4.3：一維垂直離散餘旋轉換係數矩陣。

經由一維垂直離散餘旋轉換後的矩陣，第一列的係數值通常是該行係數中最大的一個，第一列這些值稱為一維垂直離散餘旋轉換係數矩陣中的直流成分，而其他的 56 個係數則稱為一維垂直離散餘旋轉換係數矩陣中的交流成分。

$$\begin{bmatrix} 155 & 10 & 25 & -24 & -6 & 20 & -5 & -4 \\ 154 & 15 & 34 & -24 & -9 & 20 & -1 & -6 \\ 152 & 13 & 39 & -19 & -17 & 15 & 2 & -1 \\ 151 & 4 & 33 & -15 & -20 & 17 & 4 & 0 \\ 154 & 7 & 31 & -18 & -18 & 16 & 4 & -1 \\ 153 & 6 & 33 & -17 & -15 & 20 & 6 & -6 \\ 156 & 9 & 34 & -28 & -23 & 17 & 8 & -3 \\ 131 & -8 & 61 & 1 & -6 & 5 & -3 & -5 \end{bmatrix}$$

圖 4.4：一維水平離散餘旋轉換係數矩陣。

同理，經由一維水平離散餘旋轉換後的矩陣，第一行的係數值通常是該列係數中最大的一個，第一行這些值稱為一維水平離散餘旋轉換係數矩陣中的直流成分，而其他的 56 個係數則稱為一維水平離散餘旋轉換係數矩陣中的交流成分。

比較圖 4.3 和圖 4.4 的結果，可以清楚的發現當沿著正確方向作一維離散餘旋轉換時，可以有效的將影像資訊集中在直流成分中並減少交流成分的係數值。

4.2.2 水平跟垂直量化

水平跟垂直的量化矩陣是使用傳統的 JPEG 亮度量化矩陣去做修改，將傳統 JPEG 亮度量化矩陣的對角元素分別對其矩陣做列與行的延伸，即可已分別得到垂直與水平的量化矩陣。如下圖：4.5，4.6

16	16	16	16	16	16	16	16
12	12	12	12	12	12	12	12
16	16	16	16	16	16	16	16
29	29	29	29	29	29	29	29
68	68	68	68	68	68	68	68
104	104	104	104	104	104	104	104
120	120	120	120	120	120	120	120
99	99	99	99	99	99	99	99

圖 4.5：垂直量化矩陣。

16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99
16	12	16	29	68	104	120	99

圖 4.6：水平量化矩陣。

量化的公式跟傳統 JPEG 量化一樣如下：

$$\text{量化後矩陣} = \text{round} \left[\frac{\text{DCT 矩陣}}{\text{量化矩陣}} \right]$$

將圖 4.3：一維垂直離散餘旋轉換係數矩陣對圖 4.5：垂直量化矩陣做量化可得下圖 4.7。將圖 4.4：一維水平離散餘旋轉換係數矩陣對圖 4.6：水平量化矩陣做量化可得下圖 4.8。

$$\begin{bmatrix} 12 & 11 & 11 & 8 & 3 & 7 & 12 & 12 \\ 0 & 0 & 1 & 2 & 1 & -1 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

圖 4.7：一維垂直離散餘旋轉換係數量化後矩陣。

$$\begin{bmatrix} 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 9 & 0 & 2 & -1 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 10 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 8 & -1 & 4 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

圖 4.8：一維水平離散餘旋轉換係數量化後矩陣。

4.2.3 水平跟垂直係數排列

一維垂直離散餘旋轉換係數矩陣經過垂直量化表量化後，使用圖 4.7 的水平係數排列序列排序。

0	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
63	62	61	60	59	58	57	56

圖 4.9：水平係數排列序列。

一維水平離散餘旋轉換係數矩陣經過水平量化表量化後，使用圖 4.8 的垂直係數排列序列排序。

0	15	16	31	32	47	48	63
1	14	17	30	33	46	49	62
2	13	18	29	34	45	50	61
3	12	19	28	35	44	51	60
4	11	20	27	36	43	52	59
5	10	21	26	37	42	53	58
6	9	22	25	38	41	54	57
7	8	23	24	39	40	55	56

圖 4.10：垂直係數排列序列。

將圖 4.7：以圖 4.9 之水平係數排列方式重新排序後得到一維係數序列為
[SOBV 12 11 11 8 3 7 12 12 -1 0 -1 1 2 1 0 0 0 0 -1 0 0 -1 0 0 0 0 0 1 1]

其中 SOBV 表示 Start-Of-Block Vertical 的簡寫是 Huffman 編碼裡，表示一個新的 8*8 子方塊的開始，並告知在解碼的過程中該方塊重新排序跟量化表的選擇，在此的選擇是告知此方塊屬於垂直高相關性。

將圖 4.8：以圖 4.10 之垂直係數排列方式重新排序後得到一維係數序列為
[SOBH 10 10 10 9 10 10 10 8 -1 1 1 1 0 1 1 1 2 2 2 2 2 2 4 0 -1 -1 -1 -1 -1 -1 -1]

其中 SOBH 表示 Start-Of-Block Heterogeneity 的簡寫是 Enhanced Huffman 編碼裡，表示一個新的 8*8 子方塊的開始，並告知在解碼的過程中該方塊重新排序跟量化表的選擇，在此的選擇是告知此方塊屬於水平高相關性。

4.2.4 Huffman 編碼

Enhanced JPEG 中 Huffman 編碼也是建構在傳統 JPEG 壓縮上，唯一的不同在於定義 SOB(Start-Of-Block)字碼為 10100，SOBV(Start-Of-Block Vertical)字碼為 101011，SOBH(Start-Of-Block Heterogeneity)字碼為 101010，去取代 EOB 字碼 1010。定義這些字碼的目的在於給定這些重排序列相關性的屬性，SOB 表示不具水平或垂直的相關性，SOBV 表示具垂直相關性，SOBH 表示具水平相關性。

再者這三個定義碼也暗示了 DC 值的個數，由於傳統 JPEG 編碼的過程 DC 值與 AC 值係數是分開處理的因此不管是在編碼或解碼的過程，預先知道 DC 值的個數是很重要的。SOB 表示壓縮的過程使用二維的 DCT 餘旋轉換，因此重排的一維序列裡的第一個係數為 DC 值其餘的係數為 AC 值。SOBV 和 SOBH 表示壓縮的過程使用一維的 DCT 餘旋轉換，因此重排的一維序列裡前八個係數為 DC 值其餘的為 AC 值。最後傳統 JPEG 壓縮編碼裡 EOB 代表著此重排序列裡接下來的值都為零值，在 Huffman 編碼裡定義的字碼 SOB，SOBV，SOBH 代表一個 8*8 重排一維序列的開始，因此在解碼的過程中一但比對到預先定義好的這三個特殊字碼，就將前一個重排的一維序列補足 64 個數值，未填滿的部分就補足零，此步驟及可以滿足傳統 JPEG 壓縮編碼裡 EOB 意義。

Huffman 編碼的步驟如下：

1. 在每一個重排序列的開始都給定一個 Huffman 編碼表裡的特殊碼 SOB 字碼為 10100，SOBV 字碼為 101011 或 SOBH 字碼為 101010。
2. 將重排序列的 DC 值於 AC 值分開處理。
3. 將相鄰的 DC 值取其差值(Differential Pulse Code Modulation)去查表編碼。
4. 其他的 AC 值，將非零係數前面零值係數的個數及非零係數的大小類別去做查表編碼。

DC 值取其 DPCM 查表編碼與 AC 值查表編碼過程和傳統的 JPEG 壓縮編碼程序一樣，在此不再贅述。

4.3 Enhanced JPEG 解壓縮流程

Enhanced JPEG 解壓縮的過程就是將其壓縮的過程反過來。解壓縮的流程如下圖 4.11 所示：

下圖 4.11 所示：

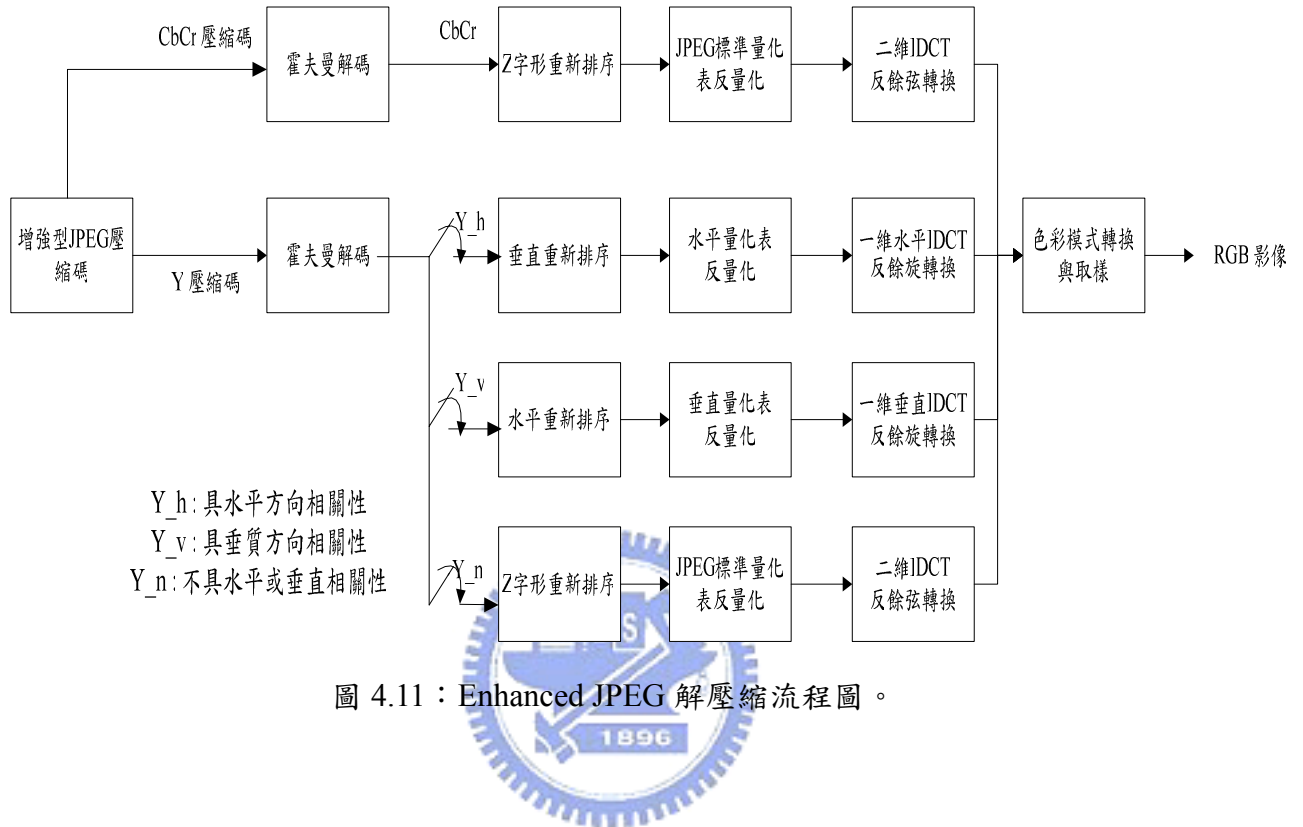


圖 4.11：Enhanced JPEG 解壓縮流程圖。

4.3.1 Huffman 解碼

解碼與編碼的過程類似需要將 DC 值跟 AC 值分開處理。解碼流程步驟如下：

1. 比對 Huffman 編碼表裡的特殊碼 SOB 字碼為 10100，SOBV 字碼為 101011 或 SOBH 字碼為 101010。SOB 有一個 DC 值而 SOBV 跟 SOBH 有八個 DC 值。
2. 根據傳統 JPEG 解碼方式先對 DC 值解碼。
3. DC 值編碼完成後在還沒出現下一個 Enhanced Huffman 編碼表裡的特殊碼時依據傳統 JPEG 對 AC 值的解碼方式解碼。
4. 比對到下一個 Huffman 編碼表裡的特殊碼時，將上一個解碼完成的重排序列未滿 64 個係數的其餘部分補上零值。
5. 重複上述 1-4 的步驟直到所有編碼的碼都被解碼完成後。

4.3.2 水平跟垂直係數還原排列

當每一個 $1*64$ 的一維係數序列被解碼完成後，將個別的一維係數序列依照個別相關性的屬性做還原排列。

水平還原排列將一維的 $1*64$ 係數序列依據圖 4.12 轉換成二維 $8*8$ 的矩陣。

0	1	2	3	4	5	6	7
15	14	13	12	11	10	9	8
16	17	18	19	20	21	22	23
31	30	29	28	27	26	25	24
32	33	34	35	36	37	38	39
47	46	45	44	43	42	41	40
48	49	50	51	52	53	54	55
63	62	61	60	59	58	57	56

圖 4.12：水平還原係數排列序列。

垂直還原排列將一維的 $1*64$ 係數序列依據圖 4.13 轉換成二維 $8*8$ 的矩陣。

0	15	16	31	32	47	48	63
1	14	17	30	33	46	49	62
2	13	18	29	34	45	50	61
3	12	19	28	35	44	51	60
4	11	20	27	36	43	52	59
5	10	21	26	37	42	53	58
6	9	22	25	38	41	54	57
7	8	23	24	39	40	55	56

圖 4.13：垂直還原係數排列序列。

4.3.3 水平跟垂直反量化

在量化過程中，將量化矩陣的每一個元素分別除以量化矩陣中相對應的係數在取其最接近的商數，現在量化的反運算便是在解碼還原後的矩陣中將每一個元素分別乘以量化矩陣中相對應的係數。

屬於垂直相關性的使用圖 4.5 之垂直量化矩陣，屬於水平相關性的使用圖 4.6 之水平量化矩陣。

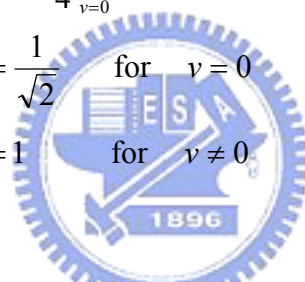
4.3.4 一維水平跟垂直 IDCT 反離散餘旋轉換

一維垂直 IDCT 反離散餘旋轉換公式如下

$$f(x, y) = \frac{1}{4} \sum_{v=0}^7 C(v) f(u, v) \cos \frac{(2v+1)y\pi}{16}$$

$$C(v) = \frac{1}{\sqrt{2}} \quad \text{for } v = 0$$

$$C(v) = 1 \quad \text{for } v \neq 0$$



8*8 的一維垂直 IDCT 轉換，將 8*8 矩陣分成八個行向量(8×1)，第一個行向量 $x = u = 1$ ，第二個行向量 $x = u = 2$ 以此類推分別引用上述的公式即可得到一個 8*8 矩陣的一維垂直 IDCT 轉換矩陣。

一維水平 IDCT 反離散餘旋轉換公式如下

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 C(u) f(u, v) \cos \frac{(2u+1)x\pi}{16}$$

$$C(u) = \frac{1}{\sqrt{2}} \quad \text{for } u = 0$$

$$C(u) = 1 \quad \text{for } u \neq 0$$

同理 8*8 的一維水平 IDCT 轉換，將 8*8 矩陣分成八個列向量(1×8)，第一個列向量 $y = v = 1$ ，第二個列向量 $y = v = 2$ 以此類推分別引用上述的公式即可得到一個 8*8 矩陣的一維水平 IDCT 轉換矩陣。

第五章 實驗結果跟比較

本章節的目地在於比較傳統數位照相機壓縮與本論文所提新形數位相機壓縮兩種方法所得到的壓縮圖片。如下圖 5.1 所示：

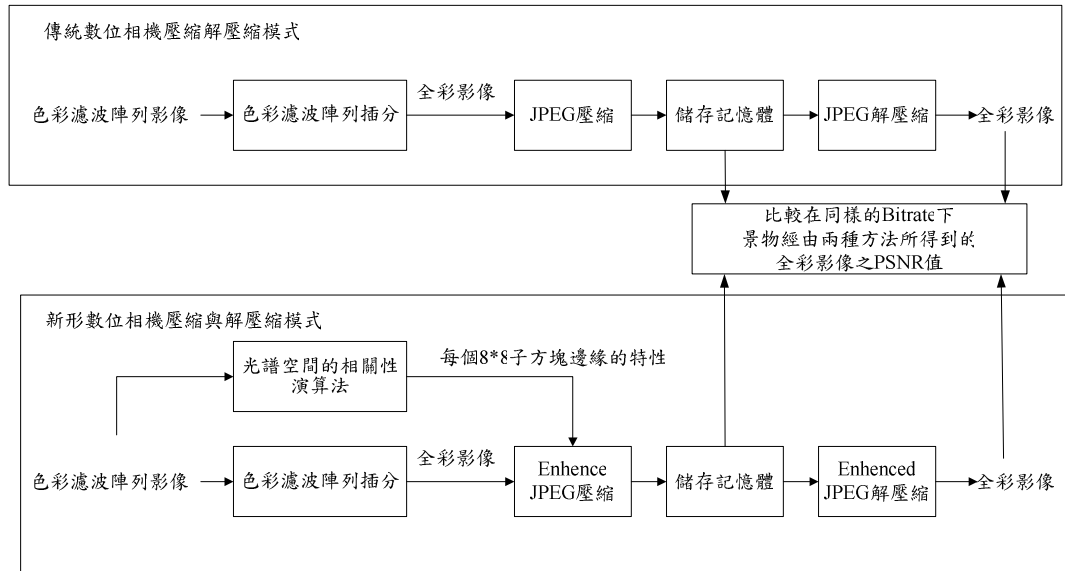


圖 5.1：傳統數位照相機壓縮與新形數位相機壓縮兩種方法比較示意圖。

圖中傳統數位相機壓縮模式中的 JPEG 壓縮，分別使用傳統 JPEG 壓縮與品質改良型 JPEG 壓縮[8]。而色彩濾波陣列插分本論文使用[13]所提出的方法，文中的 α 參數設定為 0.55。在 JPEG 與 Enhanced JPEG 壓縮中，壓縮取樣的部分本論文使用 YCbCr(4:2:0)。5.1 節中比較傳統數位相機壓縮與新形數位相機壓縮的壓縮率與 PSNR 值，其中 PSNR 在本篇論文的定義如下式：

$$PSNR(dB) = 10 \log_{10} \left\{ 255^2 \left(\frac{1}{MN} \sum_{1 \leq v \leq M} \sum_{1 \leq u \leq N} \|\bar{O}(u,v) - \bar{C}(u,v)\|^2 \right)^{-1} \right\}$$

其中 $\bar{O}(u,v)$ 表示在 (u,v) 位置上的原始彩色影像向量， $\bar{C}(u,v)$ 表示在 (u,v) 位置上原始彩色影像，經過數位相機 CFA 後經由色彩濾波插分得到全彩影像後在壓縮後的彩色影像向量。

5.2 節中就視覺上對兩種方法作進一步的比較。5.3 節綜合比較說明。

5.1 比較傳統數位相機壓縮與新形數位相機壓縮的壓縮率(Bitrate, bit/pixel)與 PSNR 值

本論文拿一般常出現的圖片如下圖 5.2 去做比較。圖片的順序由左而右由上而下共 9 張，除了圖三圖片大小為 512*512，其餘的圖片大小為 512*768。且原始圖片的 Bitrate=24(bit/pixel)。

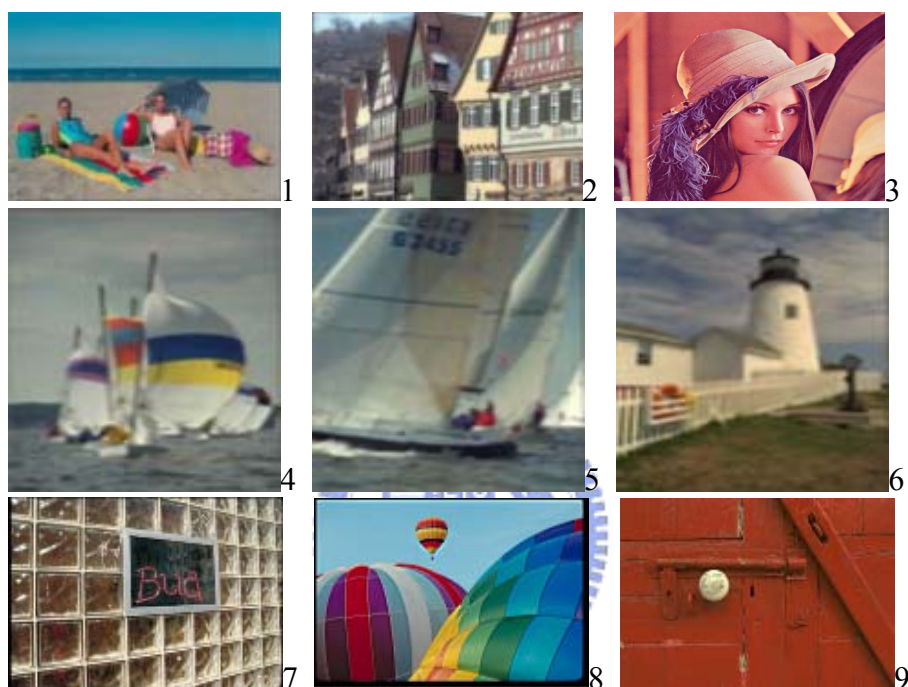


圖 5.2：實驗用的各種圖片縮小版。

對上述的 9 張圖片個別使用傳統數位相機壓縮模式與新形數位相機壓縮模式，所得的結果分別如下表 5.1，表 5.2 和表 5.3。對照表 5.1 可以得到傳統數位相機壓縮模式，使用傳統 JPEG 壓縮，在不同的量化參數下 9 張圖形個別的 Bitrate 與 PSNR 值，把 Bitrate 當 X 軸而 PSNR 值當 Y 軸可以得到下圖 5.3 中藍色的虛線(- □ -)的結果。同理對照表 5.2 可以得到傳統數位相機壓縮模式，使用傳統品質改良型 JPEG 壓縮[8]，在不同的量化參數下 9 張圖形個別的 Bitrate 與 PSNR 值，把 Bitrate 當 X 軸而 PSNR 值當 Y 軸下圖 5.3 中紫紅色虛線(- * -)。同理對照表 5.3 可以得到新形數位相機壓縮模式在不同的量化參數下 9 張圖形個別的

Bitrate 與 PSNR 值，把 Bitrate 當 X 軸而 PSNR 值當 Y 軸下圖 5.3 中紅色虛線(- -o- -)。

表 5.1：傳統數位相機壓縮模式，使用傳統 JPEG 壓縮在不同的量化參數下的壓縮率與 PSNR 值

	0.7	0.73	0.76	0.79	0.82	0.85	0.88	0.91	0.94	0.97	1
圖片 1											
bitrate	0.86492	0.84393	0.82402	0.8056	0.78796	0.77154	<u>0.75622</u>	0.74115	0.72665	0.7126	0.70017
PSNR	25.093	25.05	25.012	24.974	24.926	24.889	<u>24.851</u>	24.811	24.768	24.73	24.705
圖片 2											
Bitrate	1.5361	1.5017	1.4698	1.4404	1.4117	<u>1.3841</u>	1.3589	1.3336	1.3108	1.2871	1.2657
PSNR	24.134	24.025	23.926	23.831	23.733	<u>23.65</u>	23.559	23.478	23.392	23.309	23.225
圖片 3											
Bitrate	0.88081	0.8581	0.8373	0.8177	0.79927	0.78181	0.76542	0.7496	<u>0.73576</u>	0.72155	0.70857
PSNR	26.088	26.057	26.032	26.001	25.983	25.958	25.934	25.913	<u>25.881</u>	25.858	25.83
圖片 4											
Bitrate	0.68439	0.66757	0.65272	0.63844	0.62531	<u>0.61363</u>	0.60173	0.59124	0.5814	0.57221	0.56329
PSNR	28.755	28.68	28.61	28.538	28.478	<u>28.415</u>	28.362	28.299	28.245	28.182	28.202
圖片 5											
Bitrate	0.76748	0.74962	0.73288	<u>0.71606</u>	0.70115	0.687	0.67343	0.66161	0.64928	0.63822	0.62709
PSNR	28.77	28.687	28.593	<u>28.515</u>	28.45	28.375	28.31	28.236	28.184	28.12	28.108
圖片 6											
Bitrate	0.95663	0.93505	0.91454	0.89417	0.87502	0.85732	<u>0.84064</u>	0.82436	0.80965	0.7953	0.78143
PSNR	27.234	27.143	27.052	26.968	26.888	26.807	<u>26.725</u>	26.638	26.567	26.492	26.433
圖片 7											
Bitrate	1.2661	1.2381	1.2125	<u>1.1882</u>	1.1653	1.1436	1.1237	1.104	1.0852	1.0677	1.0503
PSNR	26.244	26.157	26.071	<u>25.986</u>	25.902	25.817	25.736	25.663	25.581	25.521	25.451
圖片 8											
Bitrate	0.51158	0.50149	0.49193	0.48367	0.47573	0.46858	0.46147	<u>0.45469</u>	0.44856	0.44278	0.43742
PSNR	28.005	27.971	27.924	27.914	27.889	27.855	27.835	<u>27.814</u>	27.782	27.76	27.801
圖片 9											
Bitrate	0.75893	0.73822	<u>0.71949</u>	0.70121	0.68397	0.66801	0.65317	0.6393	0.62554	0.61279	0.60078
PSNR	27.848	27.778	<u>27.702</u>	27.65	27.595	27.533	27.489	27.427	27.357	27.324	27.291
九張圖片平均值											
Bitrate	0.91410	0.89264	0.87280	0.85394	0.83616	0.81951	0.80385	0.78884	0.77476	0.76114	0.74831
PSNR	26.9079	26.8387	26.7691	26.7086	26.6493	26.5888	26.5334	26.4754	26.4174	26.3662	26.3384

表 5.2：傳統數位相機壓縮模式，使用品質改良型 JPEG 壓縮[8]在不同的量化參數下的壓縮率與 PSNR 值

	0.88	0.94	1.00	1.05	1.10	1.15	1.20	1.25	1.30	1.40	1.50
圖片 1											
bitrate	0.8775	0.8493	0.8220	0.8031	0.7853	0.7666	<u>0.7481</u>	0.7338	0.7213	0.6961	0.6730
PSNR	25.3000	25.2230	25.1650	25.0980	25.0470	24.9930	<u>24.9450</u>	24.9130	24.8620	24.7460	24.6740
圖片 2											
Bitrate	1.4751	1.4284	<u>1.3825</u>	1.3498	1.3188	1.2893	1.2595	1.2355	1.2134	1.1694	1.1302
PSNR	24.0780	23.9250	<u>23.7700</u>	23.6540	23.5420	23.4260	23.3200	23.2130	23.1210	22.9270	22.7350
圖片 3											
Bitrate	0.8141	0.7849	0.7580	<u>0.7379</u>	0.7191	0.7021	0.6861	0.6703	0.6573	0.6330	0.6116
PSNR	26.1950	26.1410	26.0940	<u>26.0640</u>	26.0360	25.9960	25.9670	25.9310	25.8870	25.8400	25.7800
圖片 4											
Bitrate	0.6895	0.6694	0.6505	0.6386	0.6262	<u>0.6142</u>	0.6012	0.5925	0.5841	0.5670	0.5508
PSNR	28.9230	28.8080	28.7560	28.6030	28.5260	<u>28.4140</u>	28.3860	28.3340	28.2140	28.0960	27.9550
圖片 5											
Bitrate	0.7489	<u>0.7224</u>	0.7001	0.6804	0.6679	0.6543	0.6409	0.6313	0.6147	0.5961	0.5716
PSNR	28.7850	<u>28.6610</u>	28.6070	28.4690	28.3900	28.2790	28.2120	28.1480	28.0470	27.8840	27.7850
圖片 6											
Bitrate	0.9187	0.8853	<u>0.8570</u>	0.8323	0.8153	0.7969	0.7789	0.7650	0.7456	0.7199	0.6891
PSNR	27.1730	27.0280	<u>26.9030</u>	26.7890	26.6820	26.5780	26.4920	26.3960	26.3110	26.1290	25.9820
圖片 7											
Bitrate	1.2191	<u>1.1833</u>	1.1477	1.1175	1.0948	1.0720	1.0504	1.0285	1.0121	0.9484	0.9444
PSNR	26.1870	<u>26.0340</u>	25.9200	25.7980	25.7090	25.6060	25.5050	25.4160	25.3250	25.1430	24.9690
圖片 8											
Bitrate	0.5395	0.5242	0.5206	0.5017	0.4979	0.4917	0.4847	0.4815	0.4711	0.4635	<u>0.4497</u>
PSNR	28.3170	28.2650	28.2510	28.2210	28.1620	28.1460	28.0940	28.1020	28.0430	27.9830	<u>27.9830</u>
圖片 9											
Bitrate	0.7763	0.7541	0.7297	<u>0.7112</u>	0.6926	0.6778	0.6613	0.6489	0.6294	0.6081	0.5880
PSNR	27.9620	27.8270	27.7870	<u>27.6510</u>	27.6220	27.5780	27.4790	27.4490	27.3670	27.2760	27.1770
九張圖片平均值											
Bitrate	0.8962	0.8679	0.8417	0.8200	0.8028	0.7858	0.7690	0.7550	0.7399	0.7124	0.6906
PSNR	26.9911	26.8791	26.8059	26.7052	26.6351	26.5573	26.4889	26.4336	26.3530	26.2249	26.1156

表 5.3：新形數位相機壓縮模式在不同的量化參數下的壓縮率與 PSNR 值

	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5
圖片 1											
bitrate	0.83872	0.80874	0.78565	0.76778	<u>0.75337</u>	0.74011	0.72909	0.72024	0.71242	0.70459	0.69785
PSNR	25.056	5.023	24.999	24.972	<u>24.943</u>	24.918	24.886	24.861	24.828	24.802	24.774
圖片 2											
Bitrate	1.6325	1.5484	1.4802	1.4267	<u>1.3823</u>	1.3438	1.3111	1.2834	1.2576	1.2349	1.2143
PSNR	24.351	24.202	24.069	23.962	<u>23.849</u>	23.747	23.642	23.549	23.452	23.38	23.294
圖片 3											
Bitrate	0.82697	0.8027	0.78521	0.77063	0.75939	0.74991	0.74183	<u>0.73511</u>	0.72953	0.72441	0.72007
PSNR	25.986	25.971	25.952	25.936	25.918	25.904	25.895	<u>25.879</u>	25.854	25.838	25.819
圖片 4											
Bitrate	0.74263	0.70949	0.68242	0.66134	0.64303	0.62804	<u>0.61452</u>	0.60286	0.59224	0.58257	0.57358
PSNR	29.333	29.236	29.143	29.044	28.957	28.858	<u>28.767</u>	28.674	28.567	28.497	28.397
圖片 5											
Bitrate	0.81845	0.78297	0.7559	0.73514	<u>0.71655</u>	0.70128	0.6882	0.67614	0.66575	0.65621	0.64793
PSNR	29.117	29.032	28.939	28.848	<u>28.748</u>	28.661	28.557	28.47	28.376	28.308	28.212
圖片 6											
Bitrate	0.94571	0.91091	0.88252	0.86076	<u>0.84357</u>	0.82779	0.81526	0.80416	0.79387	0.78466	0.77614
PSNR	27.238	27.175	27.117	27.057	<u>27.002</u>	26.948	26.888	26.829	26.774	26.725	26.664
圖片 7											
Bitrate	1.387	1.3136	1.2571	1.2102	<u>1.1703</u>	1.1357	1.1072	1.081	1.0571	1.0378	1.0193
PSNR	26.719	26.613	26.501	26.396	<u>26.282</u>	26.17	26.076	25.967	25.873	25.777	25.678
圖片 8											
Bitrate	0.49649	0.48439	0.47442	0.46685	0.45905	<u>0.45338</u>	0.44885	0.44544	0.44189	0.43815	0.43518
PSNR	28.079	28.061	28.034	28.02	27.994	<u>27.971</u>	27.95	27.924	27.898	27.882	27.851
圖片 9											
Bitrate	0.81516	0.78002	0.7522	0.73171	<u>0.71401</u>	0.69815	0.6857	0.67304	0.66267	0.65575	0.64689
PSNR	27.919	27.855	27.795	27.719	<u>27.672</u>	27.618	27.545	27.508	27.441	27.375	27.358
九張圖片平均值											
Bitrate	0.94485	0.90458	0.87285	0.84790	0.82684	0.80868	0.79353	0.78015	0.76812	0.75767	0.74792
PSNR	27.0887	27.0187	26.9499	26.8838	26.8183	26.7550	26.6896	26.6290	26.5626	26.5093	26.4497

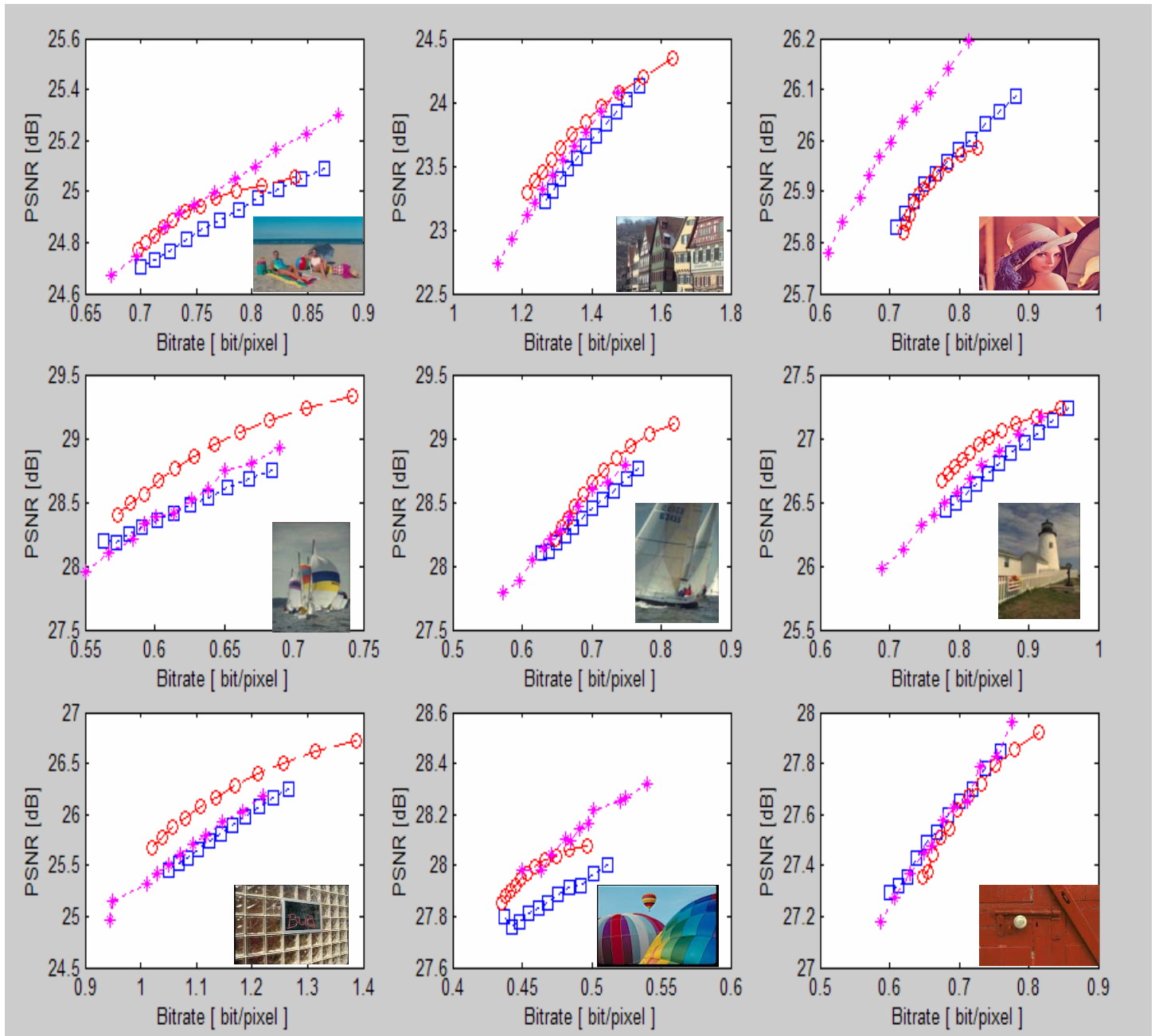


圖 5.3：圖一到圖九 Bitrate 和 PSNR 的比較，藍色的虛線(-□-)表傳統數位相機壓縮模式使用傳統 JPEG 壓縮，紫紅色虛線(-*-)表傳統數位相機壓縮模式使用品質改良 JPEG 壓縮[8]，紅色虛線(-o-)表本論文提出新形數位相機壓縮模式。

由圖 5.3 可以看出在圖片 3 和圖片 9 中傳統相機壓縮模式壓縮的效能似乎比新形相機壓縮模式要來的好，我們對此作進一步的說明。首先，先回想新形相機壓縮模式中使用 Enhanced JPEG 的方法，他的目的在於改善傳統 JPEG 壓縮圖片

中邊緣的模糊化問題。而比較圖片 3 和圖片 8，邊緣的情形較為不明顯，但是 Enhanced JPEG 在作編碼動作時對每一個 8*8 的子方塊都多用了一到兩個 bits，一張 512*768 圖片至少比傳統的 JPEG 多了 6144-12288 個 bits(傳統 JPEG 使用 EOB-1010，Enhanced JPEG 使用 SOB-10100，SOBV -101011，SOBH-101010)去告知此方塊的性質。邊緣資訊較少的影像，Enhanced JPEG 無法提供較佳的壓縮率之主因，在於平緩區域的部分都增加了一個 bit。而在這方面其實可以重新設計 Huffman 的編碼表，使得 Enhanced JPEG 不僅僅只是在邊緣較為明顯的圖片上勝過傳統的 JPEG 與品質改良型 JPEG。

下圖 5.4 為 200 張隨機抽取圖片平均值的比較，由此圖片中可以看出本論文所提出的方法有較好的影像壓縮品質。

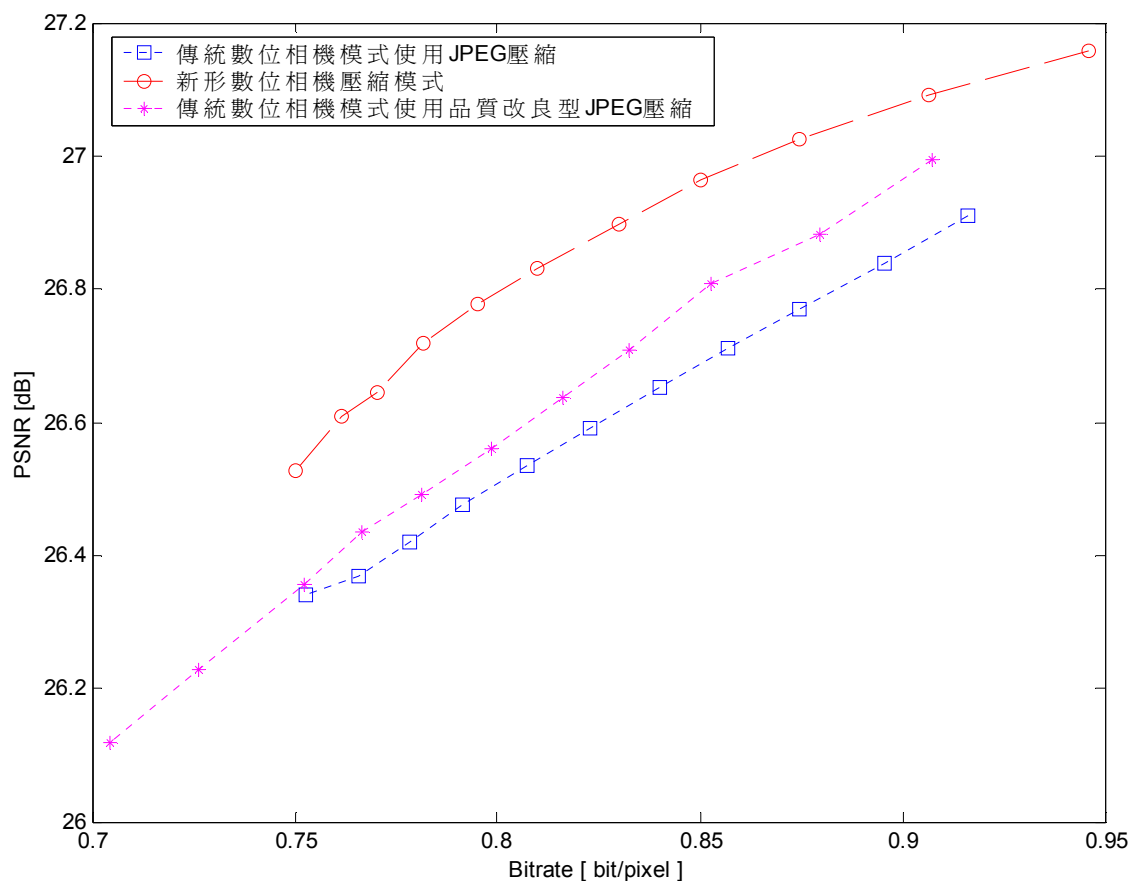


圖 5.4：200 張隨機抽取的圖片平均 Bitrate 和平均 PSNR 的比較。

5.2 傳統數位相機壓縮跟新形數位相機壓縮在視覺上的比較

將這兩個使用不同方法壓縮的圖片作進一步的比較。圖 5.5 為圖片 2 在 Bitrate 約等於 1.38(bit/pixel)下，個別用不同方法所壓縮出圖片的局部放大圖。圖 5.6 為圖片 4 在 Bitrate 約等於 0.61(bit/pixel)下，個別用不同方法所壓縮出圖片的局部放大圖。圖 5.7 和圖片 5.8 為圖片 5 在 Bitrate 約等於 0.84(bit/pixel)下，個別用不同方法所壓縮出圖片不同局部的放大圖。圖 5.8 為圖片 9 在 Bitrate 約等於 0.71(bit/pixel)下，個別用不同方法所壓縮出圖片的局部放大圖。

由下圖 5.5，5.6，5.7 和 5.8 可以明顯的發現在邊緣的部分本論文所提的方法保留了較完整的資訊，而傳統數位壓縮模式不管是使用 JPEG 壓縮方法還是改良品質 JPEG 壓縮，在邊緣的部分都模糊化了，因此我們可以知道我們所提出的壓縮方法確實可以在不增加影像大小的同時，增加了影像的品質且大幅的保留了邊緣的細節。





圖 5.5：圖片 2 視覺的細微比較 (a)表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d)本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。



圖 5.6：圖片 4 視覺的細微比較 (a)表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d)本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。

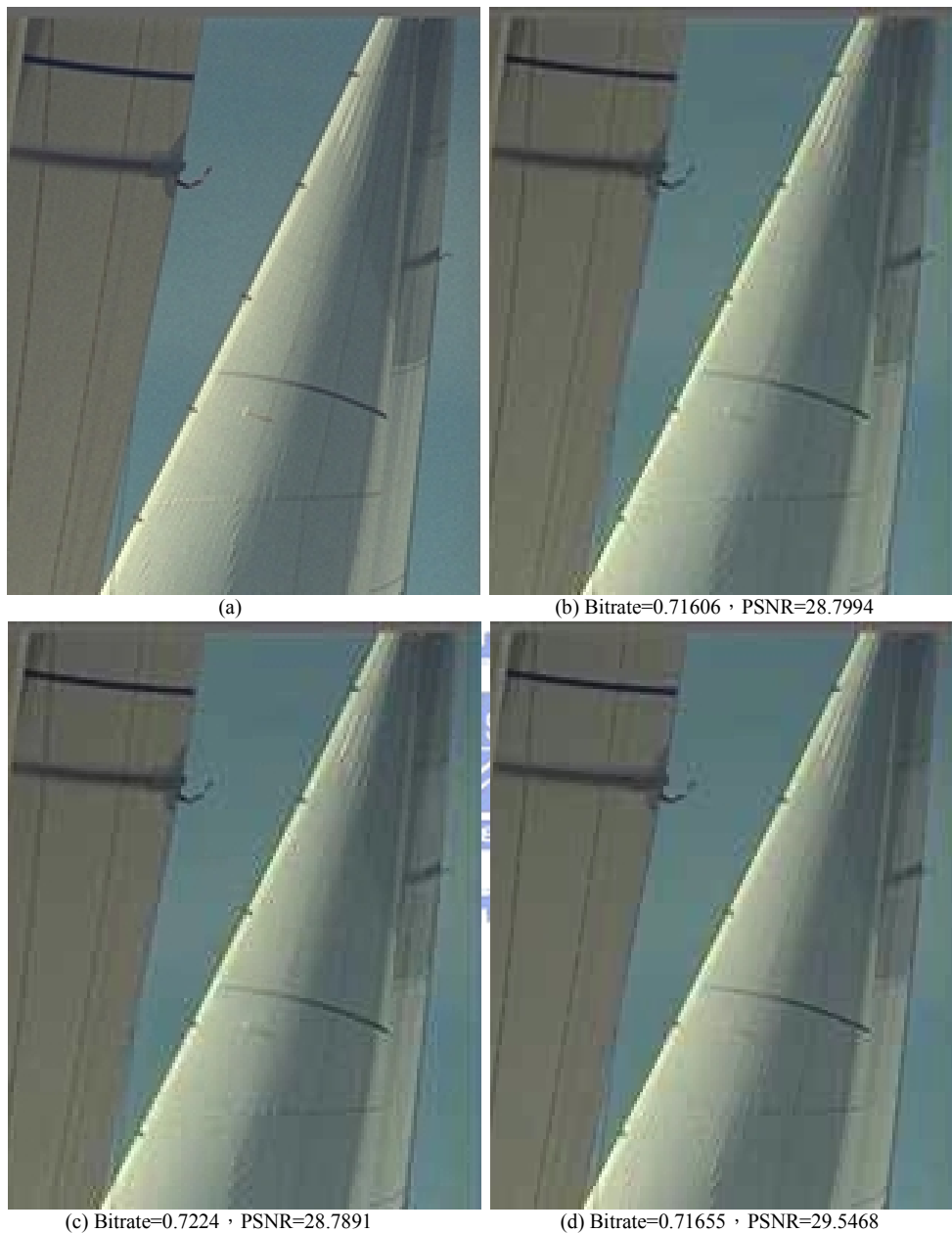


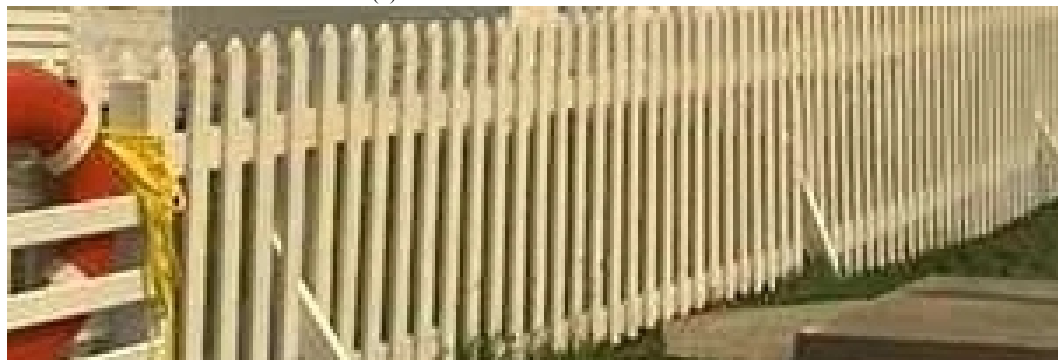
圖 5.7：圖片 5 視覺的細微比較 (a)表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d)本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。



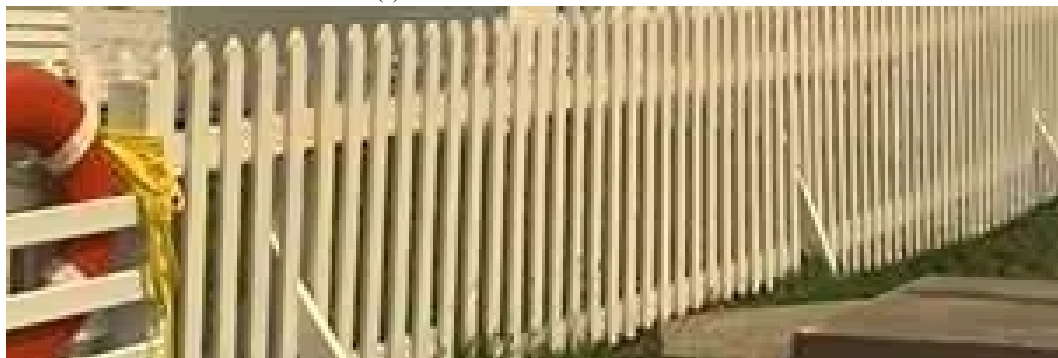
(a)



(b) Bitrate=0.84064, PSNR=24.4630



(c) Bitrate=0.8570, PSNR=24.6976



(d) Bitrate=0.84357, PSNR=25.6731

圖 5.8: 圖片 6 視覺的細微比較—(a)表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d)本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。

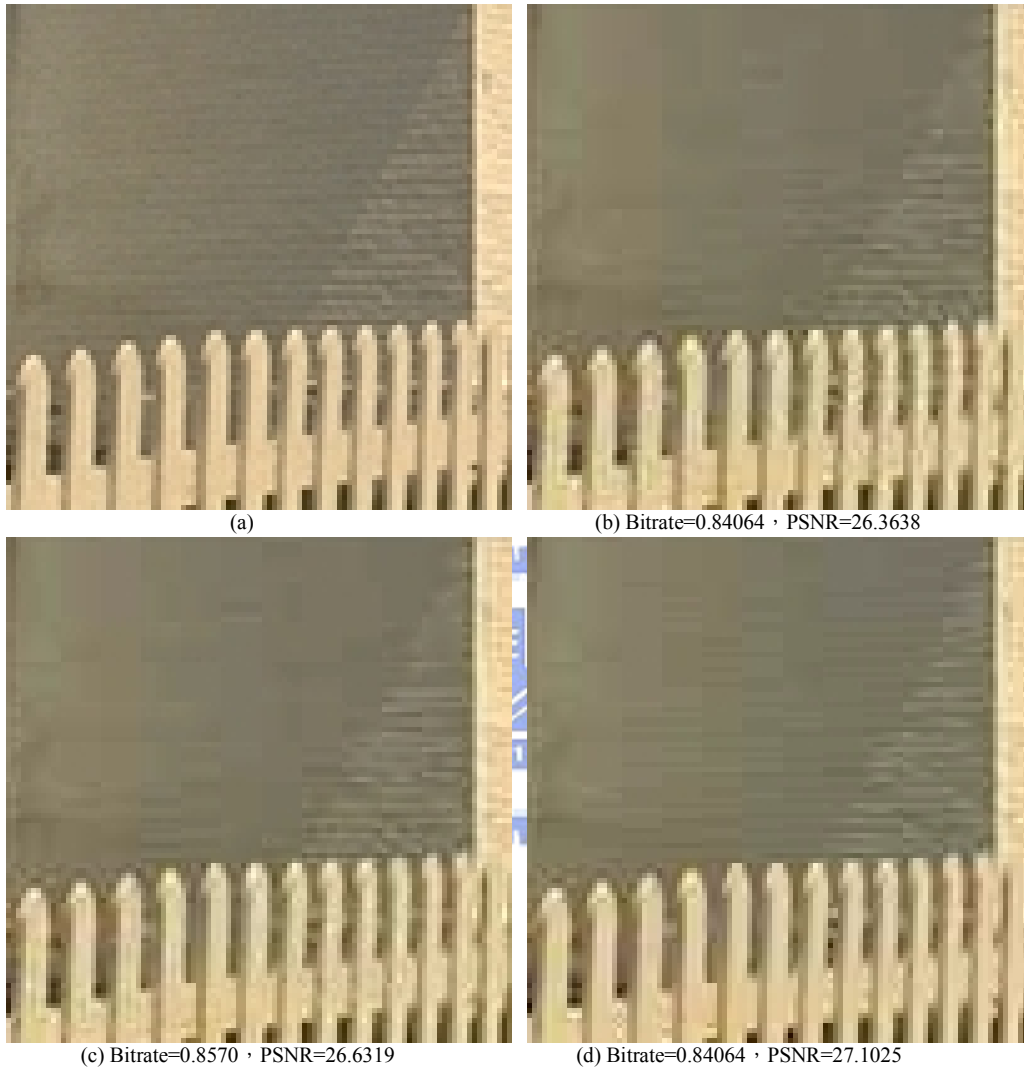


圖 5.9: 圖片 6 視覺的細微比較二(a)表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d)本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。

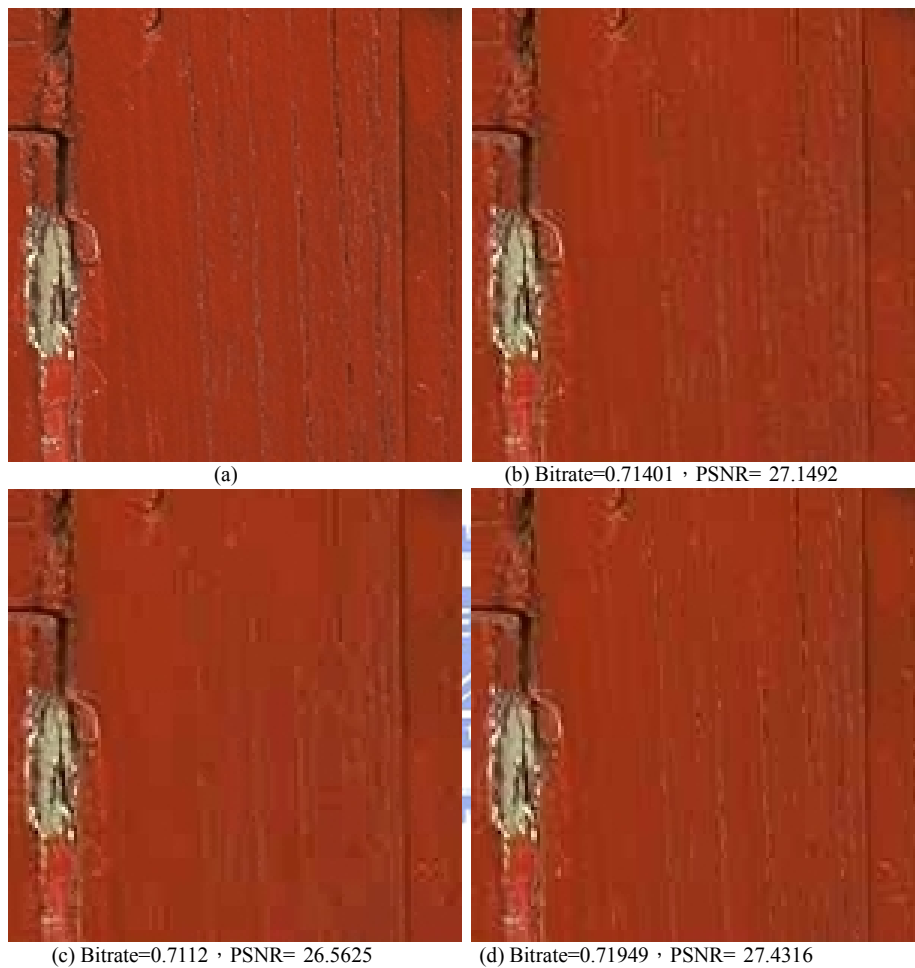


圖 5.10: 圖片 9 視覺的細微比較 (a) 表示原始影像放大圖 (b) 傳統壓縮使用 JPEG 的方法傳統壓縮(c)使用品質改良 JPEG[8] (d) 本論文所提的方法。圖中的 PSNR 值為此部分影像的 PSNR 值非整張影像。

5.3 實驗綜合比較說明

本論文所提出的壓縮方法與品質改良型 JPEG 壓縮[8]，主要都是在解決邊緣模糊化的問題。品質改良型 JPEG 壓縮主要的觀點在於記錄 JPEG 壓縮後影像與原始影像在影像邊緣亮度部分的差值影像，利用碎型(fractal)壓縮的技術將此張影像壓縮的碼加入原始 JPEG 壓縮出的碼裡。基本的架構如下圖 5.11 所示：

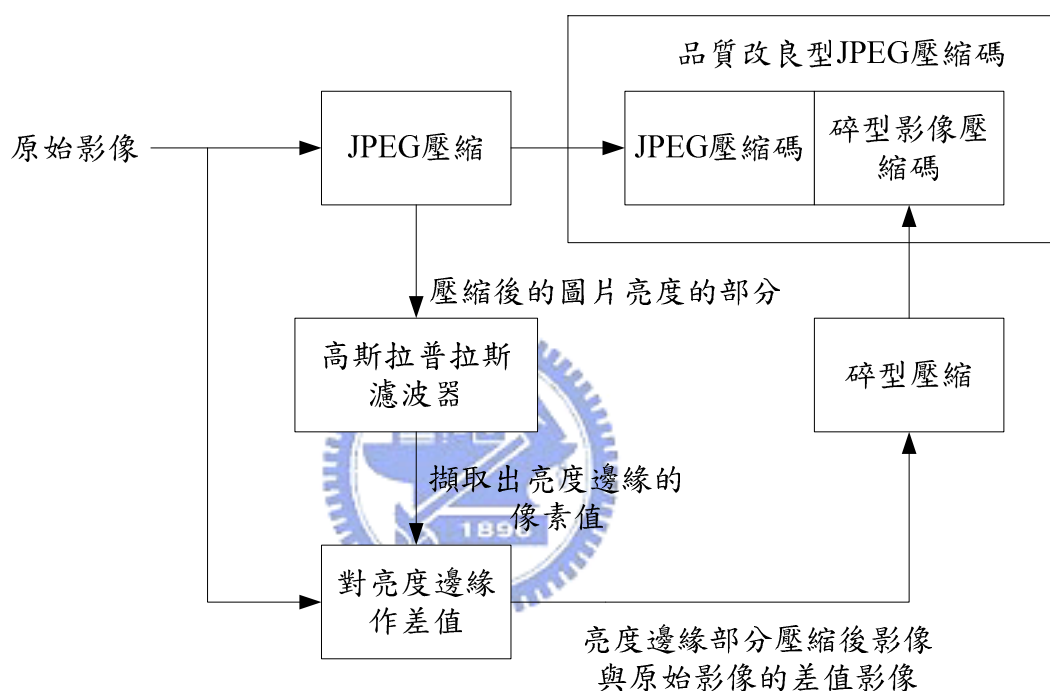


圖 5.11：品質改良型 JPEG 壓縮流程[8]。

由於碎型壓縮有很好的壓縮率與影像品質，因此這個方法利用增加些許的資料量記錄影像邊緣亮度的資訊藉此達到更好的影像品質。此方法主要的缺點如他內容裡所說需要很大的計算量，文中指出壓縮一張影像所需的時間約是傳統 JPEG 的 41 倍，這是很大的運算量更不用說在現今的數位相機裡要加入碎型壓縮的技術需要多大的成本與系統資源。反觀本論文所提的 Enhanced JPEG 壓縮方法，在影像水平跟垂直邊緣的部分，改成使用一維的 DCT 轉換藉此達到較好的壓縮率與保持較多的邊緣資訊，一維的 DCT 轉換不僅不會造成額外的運算量，

還比傳統 JPEG 使用二維 DCT 轉換的運算量要來的少。運用在現今的數位照相機裡也只需增加計算迴旋積(convolution)的能力，再修改些許的系統就可以得到更好的影像品質。

再者品質改良型 JPEG 為了補償額外來自記錄部分邊緣細節所增加的資料量，大幅的調大量化參數使得平緩區域的影像資料失真的更為嚴重，這也是為何品質改良型 JPEG 雖然整張影像的品質比傳統的 JPEG 要好，但局部放大的比較圖片品質有些甚至比傳統 JPEG 還要差的主因。舉例說明如圖片四，參照表 5.1 跟表 5.2，在相同的壓縮率下約 0.61 bitrate，傳統 JPEG 的量化參數為 0.85，品質改良型為了彌補多增加的資料量，需大幅的調高量化參數使得當量化參數為 1.15 時才可以得到相同的壓縮率由於這個理由也使得品質改良型 JPEG 壓縮技術在平緩的區域失真的更為嚴重。反觀本論文所使用的方法在平緩區域使用跟傳統 JPEG 壓縮相同的方法只改良邊緣較為明顯的部分，因此除了平緩區域的影像品質保留住之外在局部邊緣較為明顯的區域影像品質也勝優於傳統 JPEG 與品質改良型 JPEG 壓縮技術。



第六章 結論

6.1 結論

本論文所提出的 Enhanced JPEG 壓縮方法主要就是在解決邊緣模糊化的問題，傳統的 JPEG 方法不管邊緣的方向性一律對其水平與垂直做平均式的量化這點會使得邊緣附近的影像會變的較為模糊。實驗的結果印證了本論文在水平或垂直邊緣較為明顯的區域使用一維的 DCT 轉換代替二維的 DCT 轉換，可以保留較多的邊緣資訊達到更好的影像品質，在相同的 bitrate(bit/pixel)下本論文提出的方法有較好的 PSNR 值約高於傳統 JPEG 壓縮 0.3dB，這便是本篇論文的主要貢獻。

再者由於 Enhanced JPEG 主要在解決傳統 JPEG 在量化時邊緣失真導致模糊化的問題，在低失真的情況下(即量化參數約小於 0.3)時，本論文提出的方法並不能提供較好的壓縮率與 PSNR 值的主因在於，量化後的 DCT 矩陣由於量化參數較低使得高頻幾乎沒有零值，本方法主要降低字碼的主因就在於將零值有效率的集中在排序後的後段，當失去該有的零值數量時，降低字碼的主因也跟著消失，這是其中一點，再者低量化參數意味著低失真，那邊緣的模糊效果也跟著降低，本論文所想改善的部分被壓縮了，也因此改善的效果會變得較為不明顯，這是主要的第二個原因。

再者可以就運算量而言，由於在數位相機裡由馬賽克影像還原到全彩影像的過程是必要的，本論文只是將還原過程所需的每個像素邊緣的特性擷取出來，使用在壓縮的過程，也因此這部份的運算並不會因為要多計算 SSC 而增加。而在所設計的 Enhanced JPEG 壓縮藉由加入邊緣的特性，使得當每一個 8*8 方塊中若有其水平或垂直的性質只分別對此方塊作水平或垂直的一維離散 DCT 轉換，這與傳統的 JPEG 不管有否垂直或水平都作二維離散 DCT 轉算將少掉 3584 個乘法，3584 個加法。也因此本論文計算量降低的情形取決於該影像邊緣的多少，

邊緣越多時運算的時間也跟著越少，這點十分符合數位相機要求速度的原則。

6.2 後續研究

本論文提出一套新的影像壓縮方法，能將相機裡 CFA 圖片的邊緣資訊加入壓縮的過程，使得圖片有較好的壓縮品質，實驗的結果驗證了此方法的可行性，但在文中壓縮的過程水平邊緣區域與垂直邊緣區域所使用的水平與垂直量化表，文中直接拿傳統 JPEG 量化表的對角線數值作延伸，並沒有作最佳化的設計，這裡是其中一點可以再進一步改善影像品質的部分，再者由於本論文的系統必須在編碼的過程中將原本的 EOB-1010 直接改成 SOB-10100，SOBH-101010，SOBV-101011 告知每個子方塊邊緣的特性，此部分造成額外的資料量，可以重新設計針對此方法所需要用的霍夫曼編碼表，進而達到更好的壓縮率。



参考文献

- [1] J. Adams, K Parsulski, and K. Spaulking, “Color processing in digital cameras,” *IEEE Micro*, pp. 20-29, Nov.-Dec. 1998.
- [2] Y. L. Lee, H.C. Kim, and H. W. Park, “Blocking effect reduction of JPEG images by signal adaptive filtering,” *IEEE Trans. Image Processing*, vol. 7, pp. 229-234, February 1998.
- [3] Y. Yang, N. Galatsanos, and A. Katsaggelos, “Projection-based spatially adaptive reconstruction of block-transform compressed images,” *IEEE Trans. Image Processing*, vol. 4, pp. 896-908, July 1995.
- [4] Y. Yang, N. Galatsanos, and A. Katsaggelos, “Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images,” *IEEE Trans. Image Processing*, vol. 3, pp. 421-432, Dec. 1993.
- [5] Y. Kuroki, Y. Ueshige, and T. Ohta, “A new statistical model for the JPEG spatial scheme,” *Proc. IEICE*, vol. J-79-B-1, no. 7, pp. 509-511, July 1996.
- [6] Y. Kuroki, Y. Ueshige, and T. Ohta, “New statistical models of the JPEG lossless mode subject to the super high definition images,” *Proc. IEEE ICIP99*, Kobe, Japan, Oct. 1999. pp. 488-452.
- [7] Y. Kuroki, Y. Ueshige, and T. Ohta, “Redesigning of JPEG statistical model in the lossy mode fitting distribution of DCT coefficients,” in *Proc. 7th IEEE Int. Conf. Image Proc.*, Kobe, Japan, 2001, pp. 825–828.
- [8] M. Tadezawa, Y. Useshieg, K. Watanable, and M. Haswyama, “Quality improvement technique for JPEG images with fractal image coding,” in *Proc. IEEE Int. Conf. ISCAS.*, Kobe, Japan, 2005, vol. 6, pp. 6320-6323.
- [9] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression*

Standard. New York: Van Nostrand Reinhold, 1993.

- [10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. New Jersey: Prentice Hall, 2002.
- [11] M. A. Sid-Ahmed, *Image Processing*, New York: McGraw-Hall, 1994.
- [12] W. K. Pratt, *Digital Image Processing*, New York: John-Wiley, 1991.
- [13] C. Y. Tsai and K. T. Song, "Demosaicing: Heterogeneity-Projection Hard-Decision Adaptive Interpolation Using Spectral-Spatial Correlation," *Proc. SPIE*, vol. 699, pp. 1-10, 2006.
- [14] B. Bayer, "Color imaging array", U.S Patent 3971065, 1976.
- [15] S.-C. Pei and I.-K. Tan, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE trans. Circuits and Systems Video Technol.*, vol. 13, no.6, pp. 503-513, Jun. 2003.

