多媒體技術概論 AS1

105060016 謝承儒

1. DCT image compression

實作步驟

1. 先將 input image 中的 RGB 分離出來，得到三個 2-D 矩陣。
   ⇨ input_R、input_G、input_B
2. 將 input_R 和 keepRange 放入自己寫的 **DCT2** function，在裡面先從 input_R 切出 8x8 的 tmp_f，對 tmp_f 做 DCT 得到 tmp_F 和該區的 coefficient。
   DCT 的作法基於以下公式，將其轉變成矩陣運算，這是因為 matlab 對於矩陣運算有更好的表現，所以採用講義上矩陣運算的方法。(Unit2 p114,115)

$$F(u,v)=\sum_{r=0}^{M-1}\sum_{s=0}^{N-1}\frac{2C(u)C(v)}{\sqrt{MN}}f(r,s)\cos(\frac{(2r+1)u\pi}{2M})\cos(\frac{(2s+1)v\pi}{2N})$$

$$where\quad C(\delta)=\begin{cases}\sqrt{2}/2 & \delta=0\\ 1 & otherwise\end{cases}$$

3. 再將 tmp_F、keepRange 放入 **keepTheLowerFrequency** function，將 tmp_F 的高頻部分濾掉。
4. 把已經濾過高頻的 tmp_F 放進相對應的 DCT_R 的位置。
5. 接著找下一個 8x8 的 tmp_f 重複步驟 2.~4.，全部做完即可得到 DCT_R 和完整的 coefficient U_R。
6. 對 input_G、input_B 做跟 input_R 一樣的處理(步驟 2.~4)。
   ⇨ 有了 DCT_R、DCT_G、DCT_B、U_R、U_G、U_B
7. 將 DCT_R、U_R 放入自己寫的 **IDCT2** function，就可以得到 IDCT_R。
   IDCT_G、IDCT、B 也是如此得到。
   IDCT 的做法一樣參照講義上的矩陣運算(Unit2 p110)
8. 將 IDCT_R、IDCT_G、IDCT_B 組合在一起，即可得到 output
9. 將 input 和 output 放入 **computePSNR** function，就可以得到 PSNR_RGB。
   PSNR 的算法參照下方公式 ：

$$PSNR=10\cdot\log_{10}\left(\frac{MAX_I^2}{MSE}\right)=20\cdot\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

10. 把 input 放進 **RGB2YIQ**，得到 inputYIQ，接著對 inputYIQ 做跟步驟 1.~8.相同的處理，得到 outputYIQ。再將 outputYIQ 放進 **YIQ2RGB** 轉回 RGB model，接著和 input 一起放進 **computePSNR**，得到 PSNR_YIQ。
    RGB 和 YIQ 的轉換參照下方 ：

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

結果圖片

		
圖(a) 輸入的圖片	圖(b) RGB,KeepRange=2	圖(c) RGB,KeepRange=4
		
圖(d) RGB,KeepRange=8	圖(e) YIQ,KeepRange=2	圖(f) YIQ,KeepRange=4
		
圖(g) YIQ,KeepRange=8		

表 1 Q1 的結果

	RGB,KeepRange=2	RGB,KeepRange=4	RGB,KeepRange=8
PSNR_RGB	27.2944	35.6460	Inf

表 2 PSNR RGB 的值

	YIQ,KeepRange=2	YIQ,KeepRange=4	YIQ,KeepRange=8
PSNR_YIQ	27.2944	35.6460	Inf

表 3 PSNR YIQ 的值

分析以及討論

1. KeepRange 大小與 PSNR 的關係

   從圖片中可以很明顯的看出，當 KeepRange 越小、濾掉的細節越多，圖片就會越模糊。

   從 PSNR 來看，當 KeepRange 越小，PSNR 也會越低，代表 output 和 input 相差越多。

   而當 KeepRange=8，前後兩張圖是相同的，所以相當於分母 MSE=0，因此出現 PSNR = inf 的情況。

2. 由 RGB model 和 YIQ model 做處理，是否會有不同

   不管是圖片上或 PSNR，兩種 color model 都不會產生差異。

2. Dithering

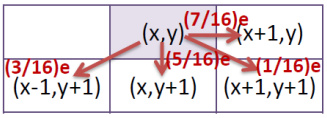
實作步驟

1. 將 input image 放入 **NoiseDithering** function。
2. 隨機產生一個 threshold。
3. 然後檢查第一個 pixel(1,1)如果高於 threshold，就為 255；反之如果低於 threshold，就為 0。
4. 選擇下一個 pixel，再次隨機產生一個 threshold，重複步驟 3.。
5. 全部 pixel 檢查完，即可得到經過 Noise Dithering 的 input image。
6. 將 input image 放入 **AverageDithering** function。
7. 計算 input image 的全部 pixel 的平均，其值為 threshold

   ⇨ threshold = average of pixels
8. 檢查所有的 pixel，如果該 pixel 高於 threshold，就為 255；反之如果低於 threshold，就為 0。結束後即可得到經過 Average Dithering 的 input image。
9. 將 input image 放入 **ErrorDiffusionDithering** function。
10. 從第一個 pixel 開始，對其值 p 做以下的檢查

$$\text{err} = \begin{cases} p, p < 128 \\ p - 255, p \geq 128 \end{cases}$$

並把 err 按照下面方法擴散



擴散完後，換到下一個 pixel，直到所有 pixel 都做過。

11. 將經過擴散後的 input image 裡的所有 pixel 做以下處理，得到經過 Error Diffusion Dithering 的 input image。

$$\text{output(i,j)} = \begin{cases} 0 & if\ input(i,j) < 128 \\ 255 & if\ input(i,j) \geq 128 \end{cases}$$

結果圖片

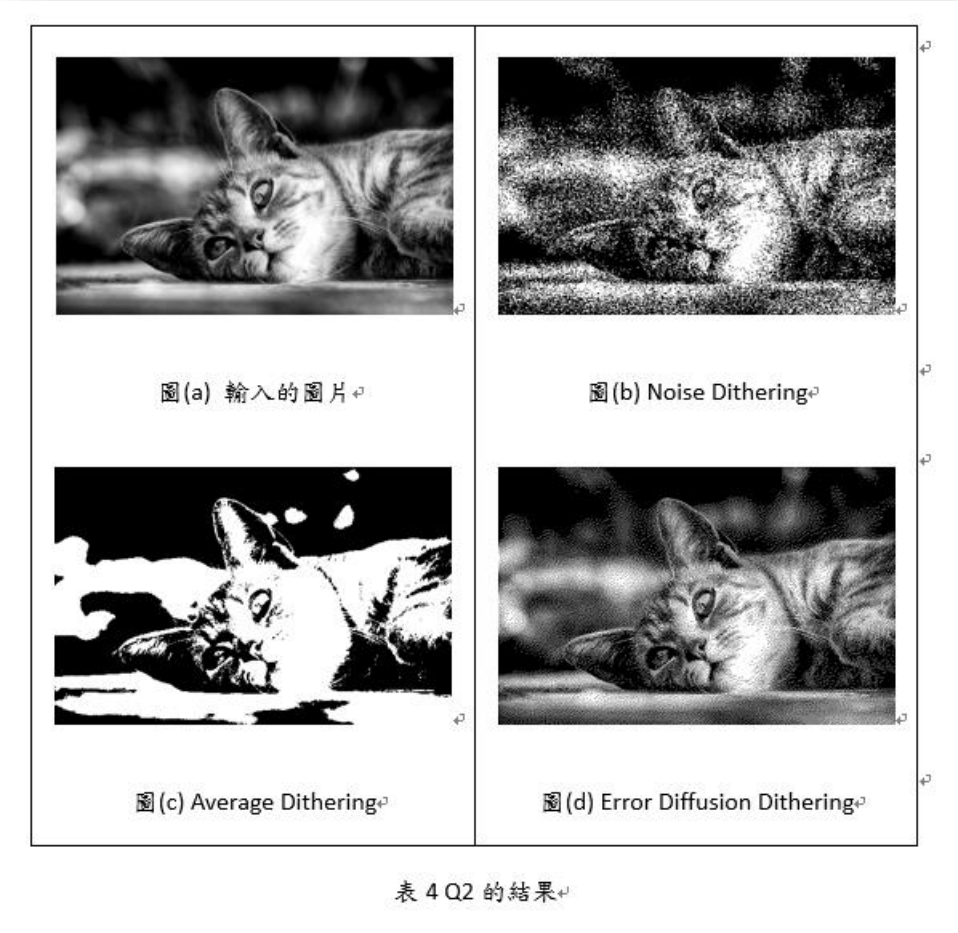
	
圖(a) 輸入的圖片	圖(b) Noise Dithering
	
圖(c) Average Dithering	圖(d) Error Diffusion Dithering

表 4 Q2 的結果

分析以及討論

1. 三種方法得出的結果比較

先將上面結果拉遠來看



圖(a) 輸入的圖片

圖(b) Noise Dithering

圖(c) Average Dithering

圖(d) Error Diffusion Dithering

表 4 Q2 的結果

經過 Noise Dithering 後的圖片，會有很類似電視機雜訊的效果，已經很明顯將原圖給遮蓋住，所以它並不是一種很有效的 Dithering 方法，但若是追求特殊效果，該方法可以使用。

經過 Average Dithering 後的圖片，會有很強烈的黑白對比。

經過 Error Diffusion Dithering 後的圖片，和原圖比起來會看起來有更多種中間顏色，且會有些許的顆粒感。

3. Image Convolution

實作步驟

1. 先將 input image 的 RGB 分成出來，得到 input_R、input_G、input_B，以及做出 Gaussian filter G。

2. 將 input_R 和 G 放入 Cov2 function。

3. 把 input_R 的四周做 zero padding，至於 padding size 就等於 G 的 size 一半

$$\text{padding size} = \left[\frac{\text{hsize}}{2}\right], [\quad] \text{是高斯符號}$$

4. 接著把經過 zero padding 的 input_R 和 G 做 convolution，得到 output_R

5. 將 input_B 和 input_G 也做跟步驟 2.~4.的處理，得到 output_G、output_B

6. 將 output_R、output_G、output_B 組合在一起，得到的 output 即是經過 Gaussian filter 的 input.

結果圖片

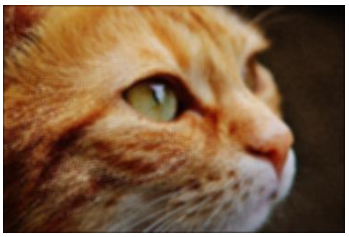
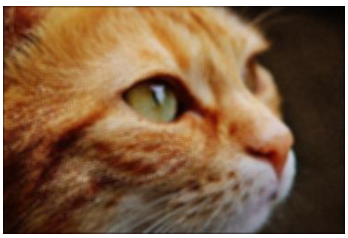
		
圖(a) 輸入的圖片	圖(b) hsize = 3x3	圖(c) hsize = 5x5
		
圖(d) hsize = 7x7	圖(e) sigma = 1	圖(f) sigma = 5
		
圖(g) sigma = 10		

表 5 Q3 的圖片

	hsize = 3x3	hsize = 5x5	hsize = 7x7
PSNR_hsize	22.5374	22.1638	22.1354

表 2 PSNR hsize 的值

	sigma = 1	sigma = 5	sigma = 10
PSNR_sigma	22.1638	20.8829	20.8500

表 3 PSNR sigma 的值

分析以及討論

1. hsize 和 Gaussian filter 的效果關係

   當 hsize 越大，越會包括越外層的 pixel，就越會讓該 pixel 自己的值比重越低，使得該 pixel 和周圍越相近，整體看起來更模糊。

2. sigma 和 Gaussian filter 的效果關係

   當 sigma 越大，filter 裡每格的比重就越均勻，就越會讓該點 pixel 和周遭的 pixel 相似，使整體看起來更模糊。