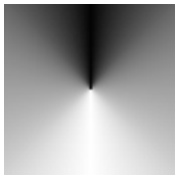
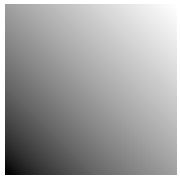
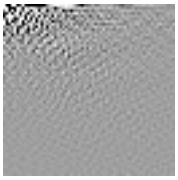
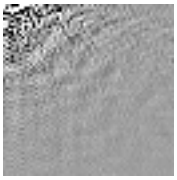
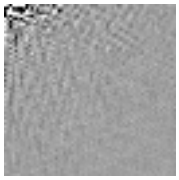
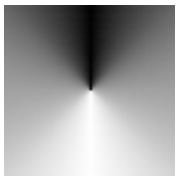
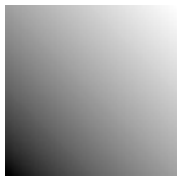
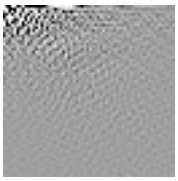
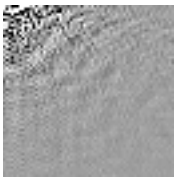
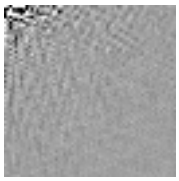
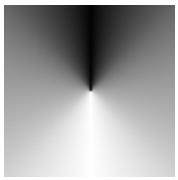
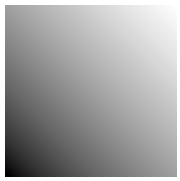
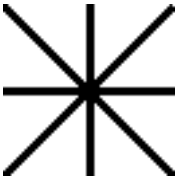
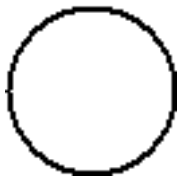
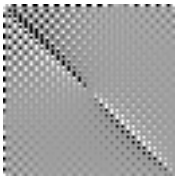
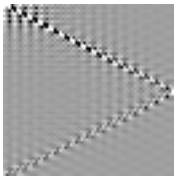
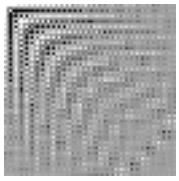
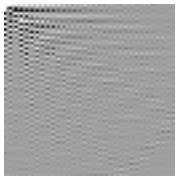
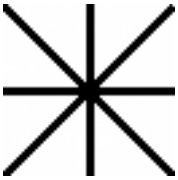
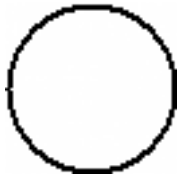
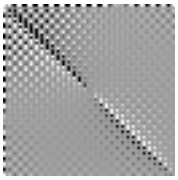
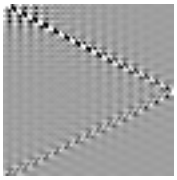
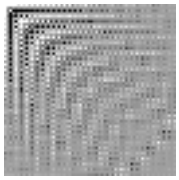
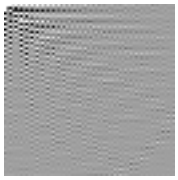
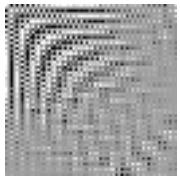
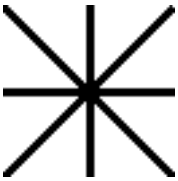
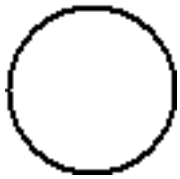
程式編譯完使用時將 argv[1]環境變數帶入要輸入的 raw 檔案，就會自動輸出 Q1-Q5（Bonus）的結果 raw 檔案。

因為 Q1-Q4 使用的 DCT&IDCT 與 Q5 的 Block-DCT&Block-IDCT 演算法相同，差別只在 Block 大小，因此 Q1-Q4 使用 Block Size 64x64 進行計算，而 Q5 則使用 Block Size 8x8 運算，只要設定 DCT&IDCT Function 第三個參數即可。

程式執行硬體環境：

1. Q1&Q2

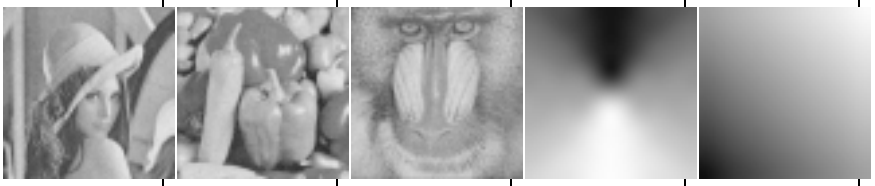
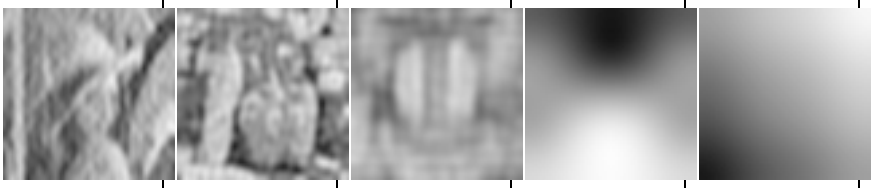
Name	Lena64	pepper64	baboon64	gra1	gra2
Origin Image					
2D-DCT					
Time	1.14205s	1.1372s	1.16301s	1.1667s	1.19388s
2D-IDCT					
Time	1.21806s	1.16037s	1.15805s	1.19827s	1.23471s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0
1D-DCT					
Time	0.013858s	0.014021s	0.013914s	0.013486s	0.015114s
1D-IDCT					
Time	0.013918s	0.017057s	0.015691s	0.01414s	0.013935s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0

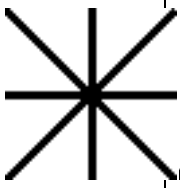
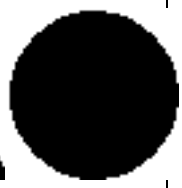
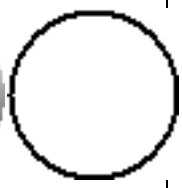
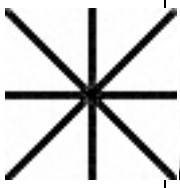
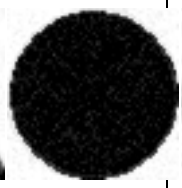
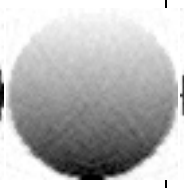
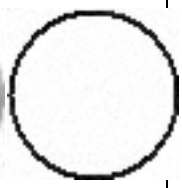
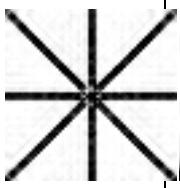
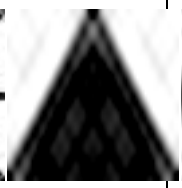
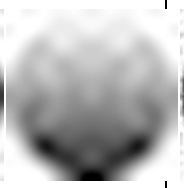
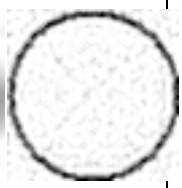
Name	wildcard	triangle	circle1	circle2	Circle3
Origin Image					
2D-DCT					
Time	1.14736s	1.15209s	1.13428s	1.14153s	1.16579s
2D-IDCT					
Time	1.19901s	1.18947s	1.16016s	1.18848s	1.17925s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0
1D-DCT					
Time	0.01503s	0.013118s	0.0138s	0.015453s	0.013674s
1D-IDCT					
Time	0.014527s	0.014479s	0.014913s	0.015861s	0.014949s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0

首先在上表可以發現 2D 的 DCT&IDCT 相較於 1D 慢，因為 2D 所需的運算量為 64*64*64*64 = 16777216 次，1D 只需要 64*64*64*2 = 524288

次，相差 32 倍的運算量。其次可以發現 MSE 的差值接近等於 0，扣除浮點數誤差，可以發現其 DCT&IDCT 轉換結果與原圖相同。對於較複雜的圖片，可以發現使用 DCT 轉為頻域空間時可以很有效的將高低頻分離。

2. Q3-Q4

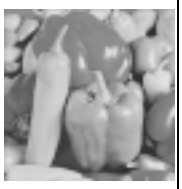
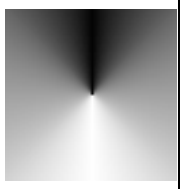
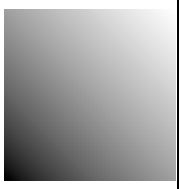
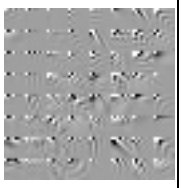
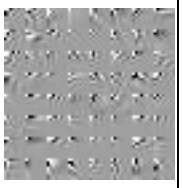
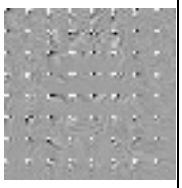
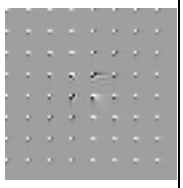
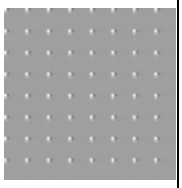
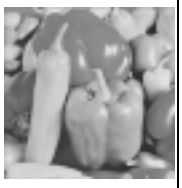
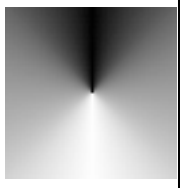
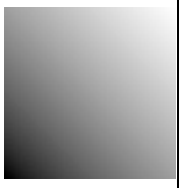
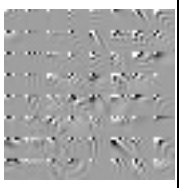
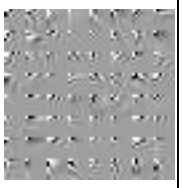
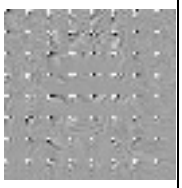
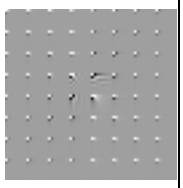
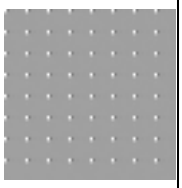
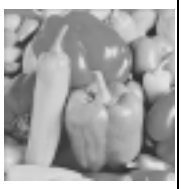
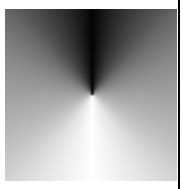
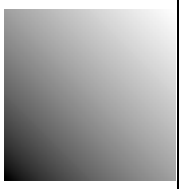
Name	Lena64	Pepper64	Baboom64	Gra1	Gra2
Origin Image					
Dead Zone	0~0	0~0	0~0	0~0	0~0
Truncate (dc,ac)	(0,3)	(0,3)	(0,3)	(0,6)	(0,5)
Quantization					
Required Bits	32767	32767	32767	32767	32767
PSNR	36.1585dB	36.172dB	36.1448dB	32.5046dB	46.2638dB
Dead Zone	-7~7	-7~7	-7~7	-7~7	-7~7
Truncate (dc,ac)	(0,3)	(0,3)	(0,3)	(0,6)	(0,5)
Quantization					
Required Bits	2007	2239	847	31	39
PSNR	24.3112dB	24.0379dB	25.5966	26.4092dB	39.8252dB

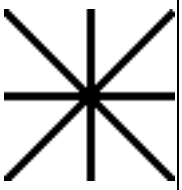
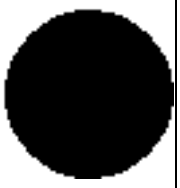
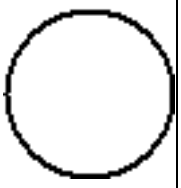
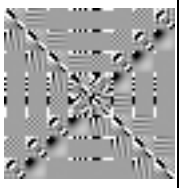
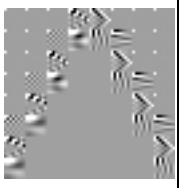
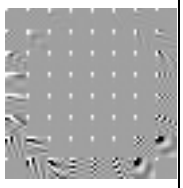
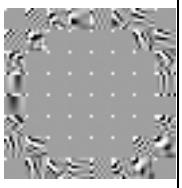
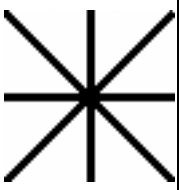
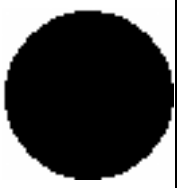
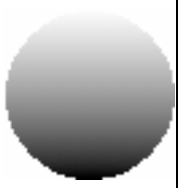
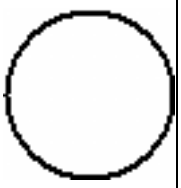
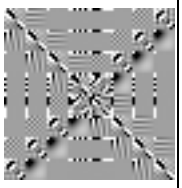
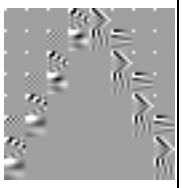
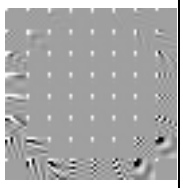
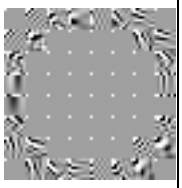
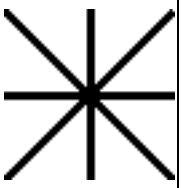
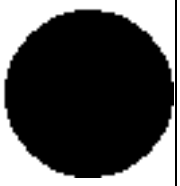
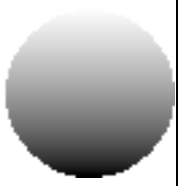
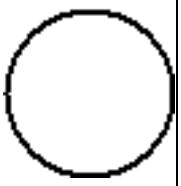
Name	wildcard	triangle	Circle1	Circle2	Circle3
Origin Image					
Dead Zone	0~0	0~0	0~0	0~0	0~0
Truncate (dc,ac)	(0,4)	(0,6)	(0,5)	(0,5)	(0,4)
Quantization					
Required Bits	32767	32767	32767	32767	32767
PSNR	32.5945dB	24.7378dB	25.8255dB	25.7357dB	31.0865dB
Dead Zone	-7~7	-7~7	-7~7	-7~7	-7~7
Truncate (dc,ac)	(0,4)	(0,6)	(0,5)	(0,5)	(0,4)
Quantization					
Required Bits	2135	183	423	231	2479
PSNR	23.3774dB	18.4531dB	16.9585	19.2867dB	17.7563dB

可以發現上表大多沒有 Dead Zone 的 Quantization 對於圖片還原後的改變 PSNR 都是在 30dB 以上，而有 Dead Zone 則會在 20dB~30dB 左右品質會下降但是 Required Bits 會變少很多，其次會發現若

是圖片本身頻率分佈均勻，則 Required Bits 則較少，因為與 DCT 轉換為頻域的方式有關。

3. Q5

Name	Lena64	Pepper64	Baboon64	Gra1	Gra2
Origin Image					
2D-DCT					
Time	0.016258s	0.017584s	0.014807s	0.016681s	0.018181s
2D-IDCT					
Time	0.016632s	0.017191s	0.016661s	0.017275s	0.017762s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0
1D-DCT					
Time	0.001765s	0.0017s	0.001953s	0.002101s	0.001694s
1D-IDCT					
Time	0.001866s	0.00189s	0.001978s	0.001701s	0.001917s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0

Name	wildcard	triangle	Circle1	Circle2	Circle3
Origin Image					
2D-DCT					
Time	0.017657s	0.018502s	0.015195s	0.015392s	0.015521s
2D-IDCT					
Time	0.019152s	0.017392s	0.017506s	0.01659s	0.017336s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0
1D-DCT					
Time	0.002056s	0.001661s	0.001707s	0.001661s	0.001864s
1D-IDCT					
Time	0.001841s	0.001665s	0.001821s	0.001662s	0.001662s
PSNR	MSE = 0	MSE = 0	MSE = 0	MSE = 0	MSE = 0

與 1.Q1&Q2 的圖表相比，使用 Block Size 8x8 速度會比直接 Block Size 64x64 快，以 2D-DCT&IDCT 為

例，64x64 要運算 64*64*64*64 = 16777216 次，而 8x8 則只要 8*8*8*8*8*8 = 262144 次，相差 64 倍，而若是 1D-DCT&IDCT 更是只要 1ms 左右的運算，因此 8x8 相較於 64x64 能夠有更好的執行時間與效率。