許主聰 資工二 B93902115 組語 Final Project

我這次的報告是以之前演算法的一個作業 DCT/IDCT 爲主，用 SSE 和 SSE2 來對它進行改良。以下是所用到的公式和量化表：

DCT:

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

IDCT:

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

$$C(u)=\begin{cases}\frac{1}{\sqrt{2}} & when \quad u=0\\ 1 & when \quad u\neq 0\end{cases}$$

$$C(v)=\begin{cases}\frac{1}{\sqrt{2}} & when \quad v=0\\ 1 & when \quad v\neq 0\end{cases}$$

量化表：

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

首先，我簡單先說一下我原來的 DCT 的作法。對於輸入一個 nxn 的矩陣而言，分割成(n/8)*(n/8)個 8x8 子矩陣，分別對每個子矩陣進行 DCT 轉換。在進行轉換之前，先建兩個重要的表。首先先觀察所用的公式

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

發現其實(1/sqrt(2N))C(u)C(u) 是只跟 u，v 有關的函數，故可以納入量化表中，但量化表必須是 float，而不是 int。

另外，對於公式中的 cos((2x+1)u*PI/(2N)) *cos((2x+1)u*PI/(2N))是跟 u,v,x,y 有關的函數，故可以對它們展開，因爲 0<=u,v,x,y<=8，故共須 8^4=4096 個空間。展開後儲存到 q[4096]中，類型爲 float。

而對一個 8x8 的矩陣而言，對它進行 row 展開，放在一個 c[64]中。接着進行 DCT 轉換，如下圖：

	0	64	128	…	4032
64 個 c 矩陣	c	c	c	…	c
q[4096]	???	????	????	…	????
對應相乘並對應每一點的 64 個數相加	第一點的總和	第二點的總和	第三點的總和	…	第 64 點的總和
除以量化表 qMatrix[64]	qMatrix[0]	qMatrix[1]	qMatrix[2]	…	qMatrix[63]
得到對應的數	k1	k2	k3	…	k64

以上是原來 DCT 的作法。

現在要對它進行改良，原理很簡單，因爲我原來的 DCT 作法已經把所要的資料排列得十分整齊，現在只要再把這些資料排在 16byte 邊界上，再利用 SSE 和 SSE2 對每四個數同時做相乘或相加就可以達到很好的效果。建表部份並沒用 SSE 和 SSE2 優化，因爲相差不是很大，爲了可讀性，把它保留，但其他部份都用 SSE 和 SSE2 優化。

至於 IDCT 和它的改良方法也是相同，只是它所建的表的內容跟 DCT 不一樣，故不再重複說明。　　(註：Debug\testData 有測試的結果)