1. 操作

點擊-滑鼠左鍵

復原-Enter

二、使用技術

1. Color to Grayscale – 用下面的公式計算灰階值，並把RGB都指派成這個顏色

gray = 0.3 \* red + 0.59 \* green + 0.11 \* blue

2. Uniform Quantization – 在0~255之間切成8份，並取頭當代表，也就是

(0、32、64、96、128、160、192、224)

看rgb落在哪一個段落就給頭的數值

Ex if RGB[green] = 31, then RGB[green]=0

3. Populosity – 先統計圖片中每種顏色出現過的次數，並排序，再從其中選

出最常使用的256種，之後再把圖片中的每個像數點的顏色跟這256種顏

色做計算(圖片的RGB參數分別跟選中的256種RGB參數分別相減後平方，之後在加起來開根號，當作誤差值)，選出與它最接近的顏色(最小的誤差值)並取代它。

4. Naive Threshold Dithering – 選取127當作閾值

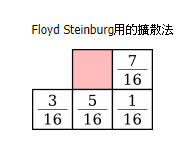
5. Random Dithering – 用rand()函數產生-51~51之間的值，將原來圖片的灰階值加上random的值，看有沒有大於127(閾值)，有就設成255否則0

6. Brightness Preserving Threshold Dithering – 先計算整張圖片的平均亮度，

之後算出需要多少個白色(RGB255)的點才能讓整體亮度為平均亮度，之後從灰階值255往下遞減，把每一點都改成255直到數超過剛剛計算出的值，之後再把其他都改成0。

7. Clustered Dithering – 用一個閾值矩陣來做計算

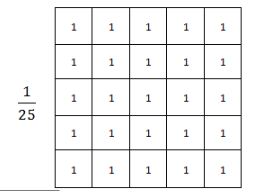
8. color floyd steinberg dithering – 利用floyd steinberg演算法把誤差擴散到周圍的點(如下圖)



其中將R分成8種顏色G分成8種顏色B分成4種顏色

9. floyd steinberg dithering – 同上的演算法，只是改用黑白兩色，並選擇127當作閾值

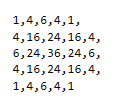
10. Box Filter – 將以自己這點為中心的5x5區域按照以下權重去做加權平均後當作自己的RGB值。



11. Bartlett Filter – 同上，只是矩陣改成



12. Gaussian Filter – 同上，只是矩陣改成



13. Half Size – 先對原本圖片做模糊處理，之後將(2i,2j)的點對應到輸出的(I,j)點

14. 先對原本圖片做模糊處理，之後將(i,j)的點對應到輸出的(2I,2j)還有(2i+1,2j+1)點