電機所-陳治言-R06921084

Description：

一開始，先處理圖片檔的標頭資訊，包括 size，offset，width，height，bits 等資訊，之後再將圖片的色彩值讀入，因為這次範例圖片只有 8 位元色彩，所以每個畫素只需要 1byte 即可儲存。而這次作業是對圖片來處理 thinning，處理完圖片之後，再將標頭與畫素資料一起寫出並存，並且產生 thinning_lena 圖片，即可完成這次的作業。

Algorithm：

一開始的 binary，使用非常簡單的迭代演算法即可，將數值小於 128 全部調整為 0，大於 128 的則是調整為 255。

再者的 marked border operator，則是透過簡單的演算法，如果偵測到四周皆有畫素，我們就將它標記為 interior，反之，則會標記成 border。

接著 pair relationship operator，則是透過與 4 連通的方式與周圍的像素標籤來做比較，如果有 interior 出現的情況，我門就將它標記起來。

最後的 connected shrink operator，透過上面的輸出，我們會得到一個標記表，透過它並與原圖做比較，決定是否要刪除此點，透過上述的步驟不停迭代，直到沒有改變發生，我們就結束迭代，別判斷最後的二極圖如果為 1，就填入 255，反之就填入 0，產生圖片並輸出。

PrincipalCode：

◎Mark_Interior_Border

```
for(i=0; i<514; ++i)
   for(j=0; j<514; ++j)
      if(binary[i][j])
         //偵測周圍，如果四周皆有畫素，標記為 2，反之為 1
         if(binary[i+1][j] && binary[i-1][j] && binary[i][j+1] && binary[i][j-1])
            bordermap[i][j] = 2;
         else
            bordermap[i][j] = 1;
      else
         bordermap[i][j] = 0;
```

◎Pair_Relationship

```
for(i=0; i<514; ++i)
    for(j=0; j<514; ++j)
        //確認該畫素須為 border，而且緊鄰 interior。
        if(1 != bordermap[i][j] || (2 != bordermap[i-1][j] && 2 != bordermap[i+1][j] && 2 !=
        bordermap[i][j-1] && 2 != bordermap[i][j+1]))
            pairmatrix[i][j] = 'q';
        else
            pairmatrix[i][j] = 'p';
```

◎Connected_Shrink

```
for(j=0; j<514; ++j)
    for(i=0; i<514; ++i)
        //當像素被標記的時候，檢查周圍像素
        if('p' == pairmatrix[i][j])
            c = 0;
            //檢查是否為可以去除的點，使用 Yokoi 四連通檢查
            if((binary[i-1][j]) && (!binary[i-1][j-1] || !binary[i][j-1])) ++c;
            if((binary[i][j+1]) && (!binary[i-1][j+1] || !binary[i-1][j])) ++c;
            if((binary[i][j-1]) && (!binary[i+1][j-1] || !binary[i+1][j])) ++c;
            if((binary[i+1][j]) && (!binary[i+1][j+1] || !binary[i][j+1])) ++c;
            if(1 == c)
                check = true;
                binary[i][j] = 0;
```

◎Main

```
//直到圖片不再發生變化，跳出迴圈
while(check)
    check = false;
    Mark_Interior_Border (bordermap, binary);
    Pair_Relationship(bordermap, pairmatrix);
    Connected_Shrink(pairmatrix, binary, &check);


for(int i=1; i<513; ++i)
    for(int j=1; j<513; ++j)
        if(binary[i][j])
            BMPdata[512-i][j].color = 255;
        else
            BMPdata[512-i][j].color = 0;
```

Parameters：

編譯程式碼 g++ -o lena lena.cpp

執行程式 ./lena lena.bmp

lena.bmp 是我們的 Input Image

ResultingImages：

thinning_lena.bmp