國立雲林科技大學

106學年度第二學期

**數位影像處理**

第一次平時作業

班 級：四電子三A

學 號：B10400001

學 生：郭鎧碩

授課教師：藍呂興 老師

繳交日期：107年04月07日

[一、 Error Diffusion 2](#_Toc510888050)

[1. 題目說明 2](#_Toc510888051)

[2. 本程式之演算法 2](#_Toc510888052)

[3. 製作方法 3](#_Toc510888053)

[4. 完整程式碼(函數寫在mask.cpp) 3](#_Toc510888054)

[5. 結果圖與原圖比較 4](#_Toc510888055)

[二、 MedianFiltering 5](#_Toc510888056)

[**1.** 題目說明 5](#_Toc510888057)

[**2.** 本程式的演算法 5](#_Toc510888058)

[3. 製作方法 5](#_Toc510888059)

[4. 完整程式碼(函數寫在mask.cpp) 6](#_Toc510888060)

[5. 結果圖與原圖比較 7](#_Toc510888061)

[三、 HistogramEqualization 8](#_Toc510888062)

[**1.** 題目說明 8](#_Toc510888063)

[**2.** 本程式的演算法 8](#_Toc510888064)

[**3.** 製作方式 9](#_Toc510888065)

[**4.** 完整程式碼(函數寫在mask.c) 10](#_Toc510888066)

[5. 結果圖與原圖比較 11](#_Toc510888067)

[四、 心得 11](#_Toc510888068)

1. Error Diffusion
2. 題目說明

為了能夠使印表機印出東西，需要把圖像轉換成只有黑(0)與白(255)。若直接將圖像做二值化，可以看到中圖色彩单调，因此出現Error Diffusion來改善。



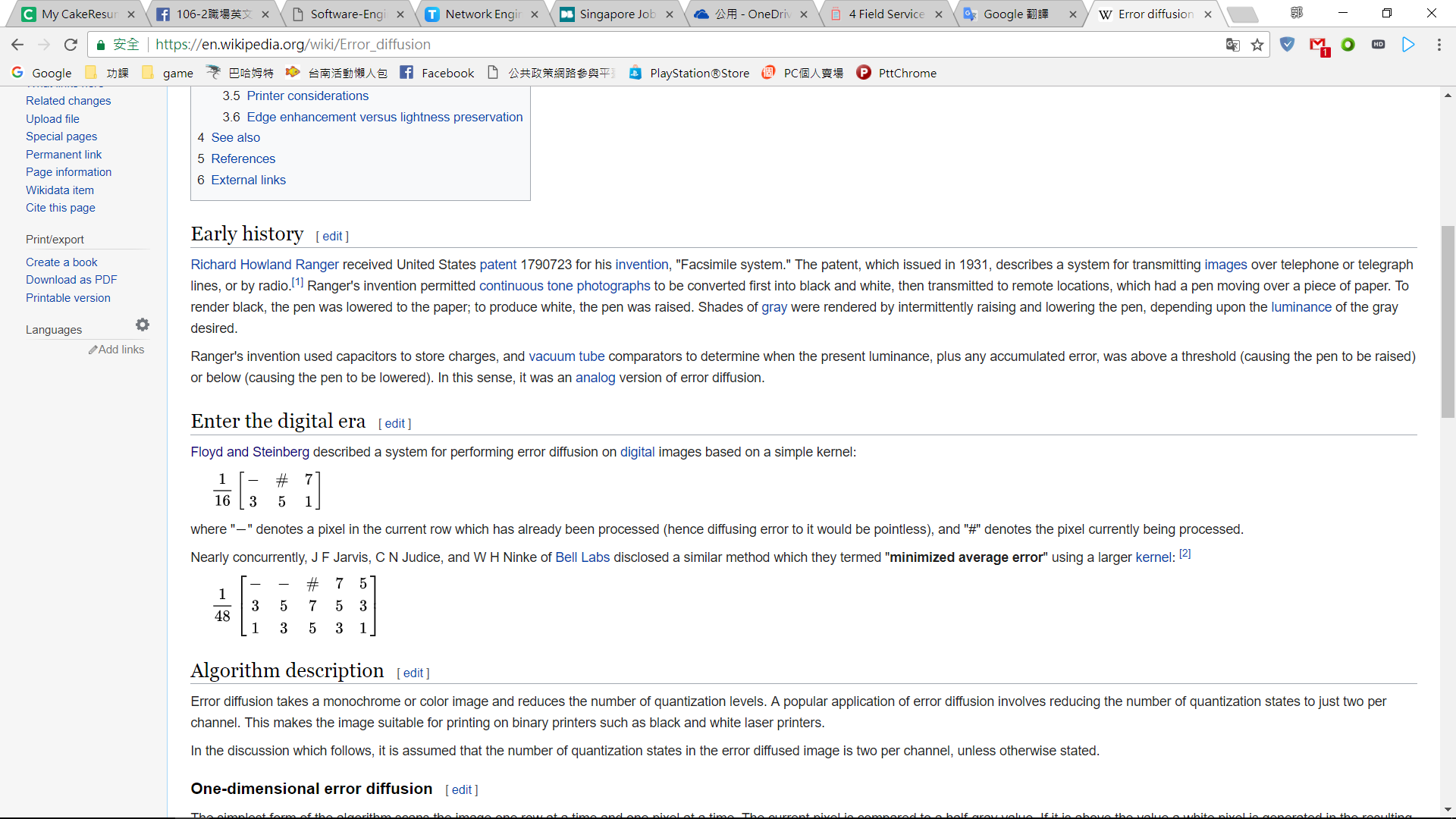
↑原圖 ↑二值化 ↑Error Diffusion

1. 本程式之演算法



1. 製作方法

透過課堂上所學到的，我們將判斷目前的像素，若大於128就轉換成225，小於則轉換成0，轉換的誤差值再擴散到還沒處理的像素，可參考下圖。一是已處理的像素，#是目前正在處理的像素，轉換的誤差值會乘以某分數擴散給其他像素



1. 完整程式碼(函數寫在mask.cpp)

Main.c

|  |
| --- |
| #include <iostream>  #include <stdlib.h>  #include "bmp.h"  #include "masks.h"  using namespace std;  int R[MaxBMPSizeX][MaxBMPSizeY];  int r[MaxBMPSizeX][MaxBMPSizeY];  int main(int argc, char \*argv[])  {  int width, height;  open\_bmp((char\*)"boat.bmp", R, R, R, width, height);  ErrorDiffusion(r, R, width, height); //import function  save\_bmp((char\*)"boat\_new.bmp", r, r, r);  printf("Job Finished!\n");  close\_bmp();  system("PAUSE");  return 0;  } |

Mask.cpp

|  |
| --- |
| #include "bmp.h"  #include <math.h>  void ErrorDiffusion(int r[1024][1024], int R[1024][1024], int& width, int& height)  {  int i, j, error;  for (i = 0; i < width; i++)  {  for (j = 0; j < height; j++)  {  r[i][j] = (R[i][j] >= 128) ? 255 : 0; //做二值化  error = R[i][j] - r[i][j]; //計算出誤差值    //將誤差值擴散  R[i + 1][j] = R[i + 1][j] + (int)round(7 \* error / 16);  if (i == 0)  continue;  else  R[i - 1][j + 1] = R[i - 1][j + 1] + (int)round(3 \* error / 16);  R[i][j + 1] = R[i][j + 1] + (int)round(5 \* error / 16);  R[i + 1][j + 1] = R[i + 1][j + 1] + (int)round(1 \* error / 16);  }  }  } |

1. 結果圖與原圖比較



1. MedianFiltering
   1. 題目說明

在圖像處理中，在進行如邊緣檢測這樣的進一步處理之前，通常需要首先進行一定程度的降噪。這個設計思想就是檢查輸入信號中的採樣並判斷它是否代表了信號。我們將使用奇數個採樣組成的觀察窗實現這項功能。觀察窗口中的數值進行排序，位於觀察窗中間的中值作為輸出。然後，丟棄原值，取得新的採樣，重複上面的計算過程。

* 1. 本程式的演算法



* 1. 製作方法

MedianFiltering的關鍵就在於讀取周圍的像素並找出中間值，透過下方的程式碼可以將半徑1的像素放入data矩陣。

|  |
| --- |
| for (x = -1;x < 2;x++)  for (y = -1;y < 2;y++)  data[3 \* (x + 1) + (y + 1)] = R[i + x][j + y]; |

再透過FindMedium函式回傳中間值，重複這些步驟即可將雜訊移除

|  |
| --- |
| int FindMedium(int data[9]) //找中間值  {  int i, j;  int temp;  for (i = 0;i < 10;i++) { //氣泡排序法  for (j = 0;j < 10;j++) {  if (data[j] < data[i]) {  temp = data[j];  data[j] = data[i];  data[i] = temp;  }  }  }  return data[4];  } |

* 1. 完整程式碼(函數寫在mask.cpp)

Main.c

|  |
| --- |
| #include <iostream>  #include <stdlib.h>  #include "bmp.h"  #include "masks.h"  using namespace std;  int R[MaxBMPSizeX][MaxBMPSizeY];  int r[MaxBMPSizeX][MaxBMPSizeY];  int main(int argc, char \*argv[])  {  int width, height;  open\_bmp((char\*)"lena\_noisy\_pepper&salt.bmp", R, R, R, width, height);  MedianFiltering(r, R, width, height); //import function  save\_bmp((char\*)"lena\_noisy\_pepper&salt.bmp", r, r, r); //第二題 printf("Job Finished!\n");  close\_bmp();  system("PAUSE");  return 0;  } |

Mask.cpp

|  |
| --- |
| #include "bmp.h"  #include <math.h>  int FindMedium(int data[9]) //找中間值  {  int i, j;  int temp;  for (i = 0;i < 10;i++) { //氣泡排序法  for (j = 0;j < 10;j++) {  if (data[j] < data[i]) {  temp = data[j];  data[j] = data[i];  data[i] = temp;  }  }  }  return data[4];  }  void MedianFiltering(int r[1024][1024], int R[1024][1024], int& width, int& height)  {  int data[9];  int x, y, i, j;  for (i = 0; i < width; i++)  {  for (j = 0; j < height; j++)  {  for (x = -1;x < 2;x++)  for (y = -1;y < 2;y++)  data[3 \* (x + 1) + (y + 1)] = R[i + x][j + y];  r[i][j] = FindMedium(data);  }  }  } |

* 1. 結果圖與原圖比較

1. HistogramEqualization
   1. 題目說明

圖像的的對比度相當接近的時候。通過這種方法，可以進行調整使圖片更容易觀看。

* 1. 本程式的演算法



* 1. 製作方式

透過下方的程式碼，我們可僅走一次圖片並取得各灰階值的個數

|  |
| --- |
| for (i = 0; i < width; i++)  for (j = 0; j < height; j++)  count[R[i][j]][1]++; |

再透過這程式碼，取得累積的個數

|  |
| --- |
| for (i = 0; i < 256; i++) {  count\_temp += count[i][1];  count\_total[i][0] = i;  count\_total[i][1] = count\_temp;  } |

最後再對每個像素做計算，即可得到新的灰階值

|  |
| --- |
| for (i = 0; i < width; i++)  for (j = 0; j < height; j++)  r[i][j] = (int)round((float)(count\_total[R[i][j]][1] - min) / (width \* height - min) \* 255); |

* 1. 完整程式碼(函數寫在mask.c)

Main.c

|  |
| --- |
| #include <iostream>  #include <stdlib.h>  #include "bmp.h"  #include "masks.h"  using namespace std;  int R[MaxBMPSizeX][MaxBMPSizeY];  int r[MaxBMPSizeX][MaxBMPSizeY];  int main(int argc, char \*argv[])  {  int width, height;  open\_bmp((char\*)"boat.bmp", R, R, R, width, height);  HistogramEqualization(r, R, width, height);  save\_bmp((char\*)"boat\_new.bmp", r, r, r);  printf("Job Finished!\n");  close\_bmp();  system("PAUSE");  return 0;  } |

Mask.c

|  |
| --- |
| #include "bmp.h"  #include <math.h>  void HistogramEqualization(int r[1024][1024], int R[1024][1024], int& width, int& height)  {  int count[256][2] = { 0 };  int i, j;  for (i = 0;i < 256;i++)  count[i][0] = i;  //計數  for (i = 0; i < width; i++)  for (j = 0; j < height; j++)  count[R[i][j]][1]++;  //累加計數  int count\_temp = 0;  int count\_total[256][2] = { 0 };  for (i = 0; i < 256; i++) {  count\_temp += count[i][1];  count\_total[i][0] = i;  count\_total[i][1] = count\_temp;  }  //找最小累加數  int min;  for (i = 0; i < 256;i++)  if (count\_total[i][1] == 0)  continue;  else {  min = count\_total[i][1];  break;  }  //計算  for (i = 0; i < width; i++)  for (j = 0; j < height; j++)  r[i][j] = (int)round((float)(count\_total[R[i][j]][1] - min) / (width \* height - min) \* 255);  } |

* 1. 結果圖與原圖比較



1. 心得

這是影像處理的第一次作業，一開始以為會很困難，像是用到指標這種很複雜的，但還好只是透過陣列做類似資料處理。因為可能以後會使用到，因此是以函數來做這次的作業，並不是用一題一個main.c檔，這樣也可以練習函數的使用方法。透過這次的作業，從上課的筆記到網路上的資料都看了很多遍，對於影像處理又有更多的認識。