影像處理與機器人視覺 HW1

姓名 : 吳建澄
學號 : NM6111035

影像處理與機器人視覺 HW1

1. 產生灰階

先將調色盤以及img array 的 function 寫好，再將datasize算一算轉成hex、little endian 的方式加進 header檔案裡，code如下圖：

```
def  RGBQUAD(QUAD_list):
    for  i  in  range(0,0xff  +  1  ,1):
        QUAD_list.append(i)  #b
        QUAD_list.append(i)  #g
        QUAD_list.append(i)  #r
        QUAD_list.append(0x00)  #reserve
```

```
def  imgarray(imglist):
    for  j  in  range(1,0xff+1):
        temp.append(j)
    temp.append(0x00)
```

```
temp.append(0x42)
temp.append(0x4D)

temp.append(0x36)
temp.append(0x0e)
temp.append(0x00)
temp.append(0x00)

temp.append(0x00)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #reserve

temp.append(0x36)
temp.append(0x04)
temp.append(0x00)
temp.append(0x00)  #offbyte

temp.append(0x28)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #bisize

temp.append(0xff)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #width  =  255

temp.append(0x0a)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #height  =  10

temp.append(0x01)
temp.append(0x00)  #plane

temp.append(0x08)
temp.append(0x00)  #bit  count
```

```
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #compression

temp.append(0x00)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #compression  size

temp.append(0x13)
temp.append(0x0b)
temp.append(0x00)
temp.append(0x00)  #horizon

temp.append(0x13)
temp.append(0x0b)
temp.append(0x00)
temp.append(0x00)  #vertical

temp.append(0x00)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #color  used

temp.append(0x00)
temp.append(0x00)
temp.append(0x00)
temp.append(0x00)  #color  important
```

寫完header檔之後，直接寫調色盤以及 for 迴圈將 image array寫入file 裡面就完成了。

```
[43]  RGBQUAD(temp)
```

```
[44]  for  k  in  range(10):
          imgarray(temp)
```

```
[45]  f=open('./grayscale.bmp','wb+')
      #  data  =  bytearray(f.read())
```

```
[46]  f.write(bytes(temp))

      3634
```

```
[47]  f.close()
```

2. 讀BMP檔

利用python open函式即可打開 file，再把它append到list裡面即可存取裡面的byte資料

```
with  open("testgray.bmp",  "rb")  as  f:
        byte  =  f.read(1)
        while  byte  !=  b"":
                #  Do  stuff  with  byte.
                #  print(byte)
                bk_list.append(byte)
                byte  =  f.read(1)
```

```
print(bk_list[:])
```

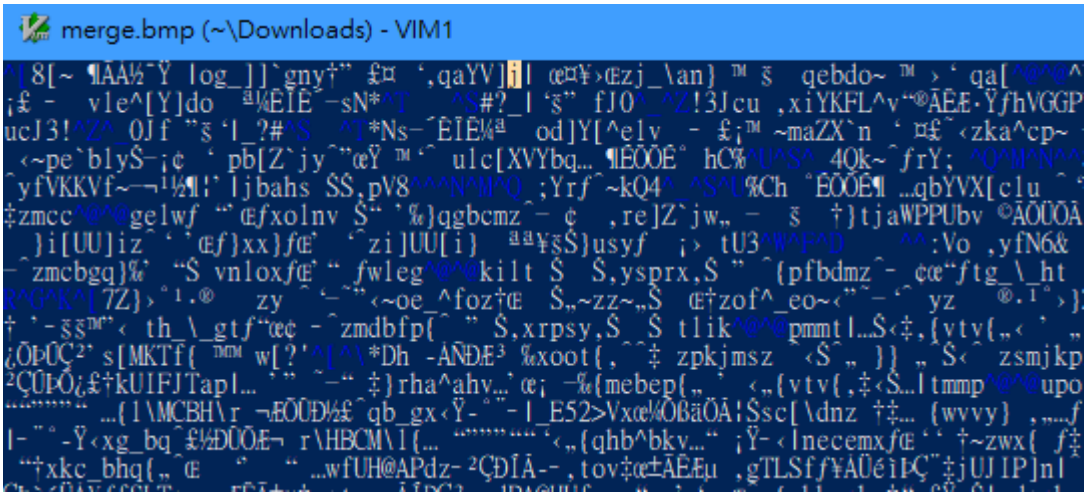
```
[b'B', b'M', b'N', b'\xfa', b'\x00', b'\x00',
```

3. 將讀進來的BMP檔頭資訊show出來

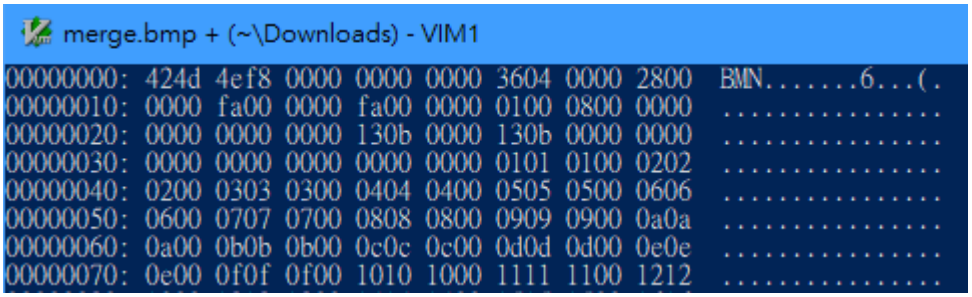
利用vim即可查看BMP header imfomation，步驟如下：

開啟commend prompt 輸入: “vim {path}”

一開始會呈現亂碼



再輸入: “:%! xxd” 即可正常顯示 header info



3. 影像處理

3-1. background

先定義中間兩點的位置

```
x1 = 250 // 2 #85
y1 = 250 // 3 #128
x2 = 250 // 2
y2 = 250 // 3 * 2
```

再利用內建函式math.cos 以及圓的公式來計算震幅, 以取得一個圓的 function

```
def calgray(Xorg,Yorg,Xtar,Ytar):
    R = (float(Xorg) - float(Xtar)) ** 2 + (float(Yorg) - float(Ytar)) ** 2
    R = math.sqrt(R) # Distance
    # Distance / 2 pi
    res = math.cos(R*math.pi/180 * 24)* 127.5 + 127.5
    return res
```

將以及兩個圓相加除以 2 以得到最後的點波源干涉

```
for i in range(250):
    for j in range(250):
        k = calgray(x1,y1,i,j)
        k2 = calgray(x2,y2,i,j)
        ksum = int((k + k2) / 2)
        # if(ksum == 256):
        #     ksum = 255
        img.append(ksum)
    img.append(0x00)
    img.append(0x00)
    img2.append(img)
    del img
    img = []
```

最後再將header, RGB調色盤, 點波源干涉整併寫進 file 就完成了

```
RGBQUAD()
```

```
[16] for k in range(250):
         for m in range(252):
             temp.append(img2[k][m])
```

```
[17] f=open('./dot.bmp','wb+')
     # data = bytearray(f.read())
```

```
[18] f.write(bytes(temp))

     64078
```

```
[19] f.close()
```

3-2. normalization & combine

先將機器人的圖以及點波源圖分別讀進來，再將他們的img array分割出來轉為 int

```
with open("testgray.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
                # Do stuff with byte.
                bk_list.append(byte)
                byte = f.read(1)
```

```
with open("bk.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
                # Do stuff with byte.
                dot_list.append(byte)
                byte = f.read(1)
```

```
for i in range(1078,len(dot_list)):
    dot_int.append(int.from_bytes(dot_list[i], "big"))
    bk_int.append(int.from_bytes(bk_list[i], "big"))
```

再將他們的權重分配好，做max normalize 到0~255之間

```
temp = []
for i in range(len(bk_int)):
    temp.append( weight * bk_int[i] + dot_int[i])
```

```
MAX = max(temp)
```

```
for i in range(len(bk_int)):
    normolize.append(int(( weight * bk_int[i] + dot_int[i]) / MAX) * 255))
```

最後一樣將Header、RGB調色盤以及normalize後的list整併在一起寫進file

```
file_merge = open("./merge.bmp","wb")
```

```
file_merge.write(bytes(normolize))
```

```
64078
```

```
file_merge.close
```

merge.bmp是結合助教給的點波源圖，merge1.bmp是結合自己產生的點波源圖

3-3 顏色改變

inverse須將原始圖檔調色盤從 0 to 255, 改寫成 255 to 0即可

```
img_list = []
header_byte = 54
inv = 255
RGBQUAD = 256*4
```

```
with open("testgray.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
                # Do stuff with byte.
                img_list.append(byte)
                inv_list.append(byte)
                byte = f.read(1)
```

```
file_blue = open("./blue.bmp", "wb")
```

```
for i in range(header_byte,header_byte + RGBQUAD):
        img_list[i] = inv.to_bytes(1, 'big') #b
        if((i - header_byte) % 4 == 3): #bule %
            img_list[i] = b'\x00'
            inv -= 1
    # print(bk_list[i])
```

```
for i in range(len(inv_list)):
    file_blue.write(img_list[i])
```

```
file_blue.close()
```

```
img_list = []
header_byte = 54
RGBQUAD = 256*4
```

```
with open("testgray.bmp", "rb") as f:
        byte = f.read(1)
        while byte != b"":
                # Do stuff with byte.
                img_list.append(byte)
                byte = f.read(1)
```

```
file_blue = open("./blue.bmp", "wb")
```

```
for i in range(header_byte,header_byte + RGBQUAD):
    if((i - header_byte) % 4 == 2): #bule %4 == 0, green %4 == 1, red %4 == 2
        img_list[i] = b'\xff'
    if((i - header_byte) % 4 == 1): #bule %4 == 0, green %4 == 1, red %4 == 2
        img_list[i] = b'\xff'
    # print(bk_list[i])
```

```
for i in range(len(img_list)):
    file_blue.write(img_list[i])
```

```
file_blue.close()
```

顏色改變只需要將inverse檔案的調色盤做更改就可以，紅色色調改藍綠色調色盤改成0x00，綠色色調改藍紅色調色盤改成0x00即可，黃色則是將綠色變成0x00