

成 功 大 學
系 統 及 船 舶 機 電 工 程 學 系

數位影像處理

Project 2

授課老師：江珮如

系 級：系統 112

學 生：周呈陽

學 號：F14081046

中華民國一百一十一年五月二十二日

流程圖



流程說明

1. 邊緣偵測

先讀取 2-X 的任意一張圖，從反覆的測試發現 RGB 中，原圖中維度 G 經 **aproxycanny** 邊緣化，並將閾值設為 **0.06** 其效果最好，生成一個邏輯矩陣 BW。

2. 判斷晶格四周

生成兩個與原圖大小相同的零矩陣(gr 和 re)，跑兩個迴圈，第一個迴圈判斷 BW 的縱向的密度，當縱向密度大於 $1942/6$ 時，將這一行的值讀進 gr 矩陣對應的位置裡；反之，不讀取其值。第二個迴圈判斷 BW 的橫向的密度，當縱向密度大於 $1942/8$ 時，將這一系列的值讀進 re 矩陣對應的位置裡；反之，不讀取其值。並可以透過 re

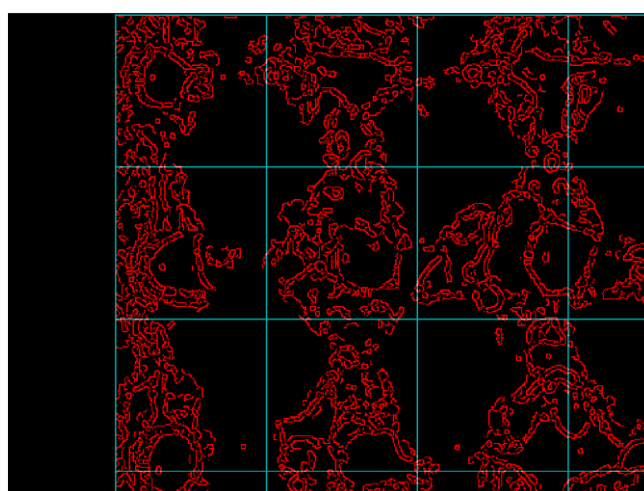
和 gr 兩個矩陣，找出晶格結構的邊界，如下圖。



圖(一)、圖(二) 判斷晶格邊界

3. 找出晶格頂點，分割 16x16 格

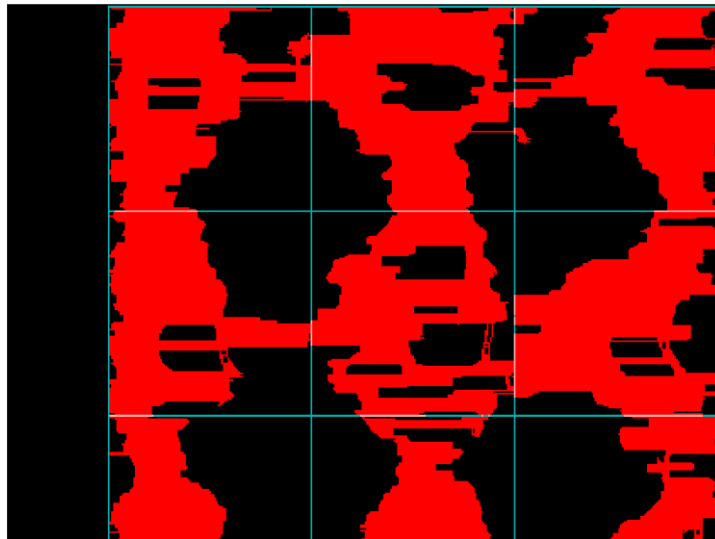
透過晶格邊界，我們可以找到晶格邊界的四個座標，並透過算出晶格的長和寬，再將原圖分隔成 16x16 個晶格，2-2 和 2-4 的照片皆為 16x16 個晶格所組成，2-1 的照片為 13x13，2-3 的照片則為 21x21，但經過多次除錯發現，切隔成 16x16 的效果會最好，故將程式碼寫切成 16x16 個晶格，下圖為左上角的部分截圖。



圖(三) 16x16 晶格的部分截圖

4. 晶格反射陰影補色

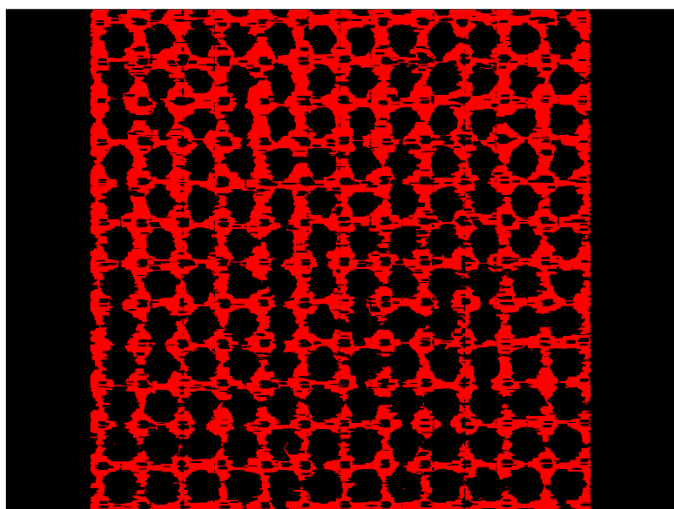
透過分割後的每一個小晶格，將其白色邊界內連線，目的為了補滿晶格中反射的陰影區塊，透過迴圈判斷，當第一個值為白色，後面接續 30 個值有白色，將第二個值也宣告為白色，從左至右，從右至左，共進行兩次；再從上至下，進行一次，下圖為左上角補色的成果。



圖(四) 16x16 晶格補色後的部分截圖

5. 建立遮罩，疊圖輸出

生成一個與原圖大小相同的零矩陣作為遮罩，將補滿陰暗處後的晶格依照原圖位置建立遮罩，下圖為遮罩，再將原本的照片使用 `imlincomb` 疊加遮罩到原圖上，`imresize` 成 1/4 原圖大小，並用 `imwrite` 輸出圖片。



圖(五) 遮罩

6. 建立函式

最後再寫成函數 `pj2new(img, name)`，`img` 為輸入的照片，`name` 可自行輸入，輸出「`pj2_ + name`」的 `jpg` 圖檔。

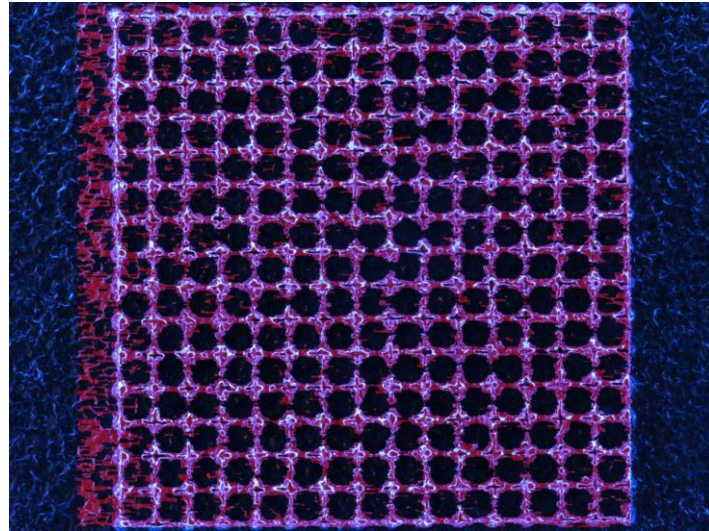
結果分析

這次的 Project 雖然可以針對不同照片調整不同的參數，但我希望能以不改變任何參數的情況下，使用同一個程式碼完成這次的 Project，也因此會與理想的結果有些差異，以下是我認為造成誤差的三個原因，原因如下：

1. 誤判晶格邊界誤判

我透過邊緣化後的密度，去判斷晶格邊界，但有部照片晶格邊界外的雜訊密度高於晶格邊界，也因此會導

致誤判邊界，同時因為程式碼的設計上無參數的調整，我僅針對部分嚴重誤判晶格邊界的圖片，進行部分修正，但距離真正的邊界還有一小段誤差，如下圖所示。



圖(六) 誤判左側邊界

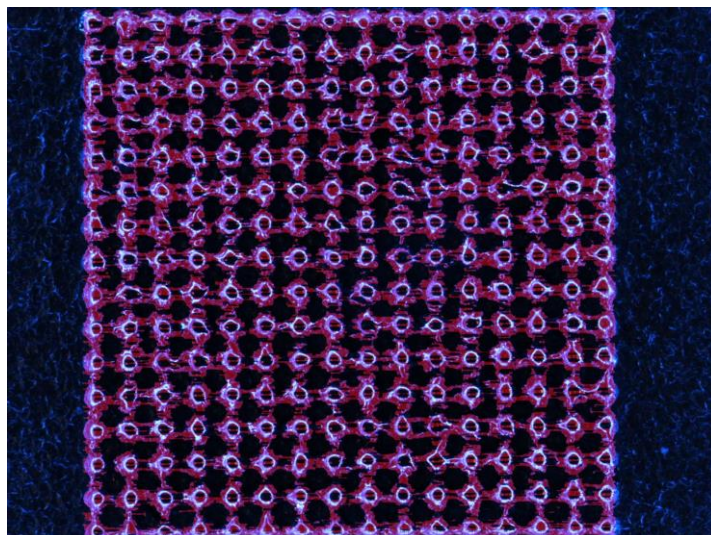
2. 晶格間隔數不同

因為程式碼的設計上無參數的調整，2-1~2-4 照片中相鄰晶格間的陰暗處數量不一，誠如流程說明的第 3 點，因為切割時會有誤差，導致無法精確的切割晶格，也無法補滿晶格上反射部分；同時也會因切割位置不同，導致誤判晶格間的陰暗處為反射的黑色部分，進而填補陰暗處，故造成結果的誤差。

3. 反射陰暗處過大

晶格上反射的陰影部分，若陰影部分過大，會導致

無法完全用遮罩補滿，進而降低結果的準確度，如下圖所示。



圖(七) 晶格反射陰影過大

程式碼

```
function pj2new(img,name)
g2=img(:,:,2);
BW = edge(g2,'approxcanny',0.06);
[a,b]=size(BW);
gr=zeros(a,b,3);
re=zeros(a,b,3);
for y=1:b
    if sum(BW(:,y))>(1942/6)
        gr(:,y,2)=BW(:,y);
    end
end
for x=1:a
    if sum(BW(x,:))>(2590/8)
        re(x,:,1)=BW(x,:);
    end
end
```



```
left=zeros(1,a)+500;
for i=1:a
    for j=1:b
        if gr(i,j,2)==1
            left(i)=j;
            break
        end
    end
end
left_val=min(left);
if left_val<250
    left_val=250;
end
right=zeros(1,a);
for i=1:a
    for j=b:-1:1
        if gr(i,j,2)==1
            right(i)=j;
            break
        end
    end
end
right_val=max(right) ;
if right_val>2350
    right_val=2350;
end
up=zeros(1,b)+500;
for j=1:b
    for i=1:a
        if re(i,j,1)==1
            up(j)=i;
            break
        end
    end
end
up_val=min(up);
down=zeros(1,b);
for j=1:b
```



```

for i=a:-1:1
    if re(i,j,1)==1
        down(j)=i;
        break
    end
end
end
down_val=max(down) ;
col=floor((down_val-up_val)/16);
row=floor((right_val-left_val)/16);
z=zeros(a,b,3);
for i=1:16
    for j=1:16
        z(up_val+col*(i-1)+1:up_val+col*i+1,left_val+row*(j-1)+1:left_val+row*j+1,1)=BW(up_val+col*(i-1):up_val+col*i, ...
            left_val+row*(j-1):left_val+row*j);
        for k=up_val+col*(i-1)+1:up_val+col*i+1
            for l=left_val+row*(j-1)+1:left_val+row*j+1
                if (z(k,l,1))==1
                    if (sum(z(k,l+1:l+30,1))>1)
                        z(k,l:l+1,1)=1;
                    end
                end
            end
        end
        for k=up_val+col*(i-1)+1:up_val+col*i+1
            for l=left_val+row*(j)+1:-1:left_val+row*(j-1)+1
                if (z(k,l,1))==1
                    if (sum(z(k,l-30:l-1,1))>1)
                        z(k,l-1:l,1)=1;
                    end
                end
            end
        end
    end

    for k=up_val+col*(i-1)+1:up_val+col*i+1-5
        for l=left_val+row*(j-1)+1:left_val+row*j+1
            if (z(k,l,1))==1

```

```

        if (sum(z(k+1:k+5,1,1))>1)
            z(k:k+1,1,1)=1;
        end
    end
end
end
end

end
end
bwnew=zeros(a,b,3);
bwnew(:,:,1)=z(:,:,1)/1.5;
KK=im2double(img);
K3 = imlincomb(1,KK,0.7,bwnew);
K4=imresize(K3,0.5);
imwrite(K4,sprintf('pj2_%d.jpg',name))
end

```

(壓縮檔內含兩種程式碼:pj2new 為函式、run_new 為執行檔)

心得

這次的 Project2 難度直接大幅提升，也慶幸我很早就開始思考怎麼做，其實網路上找不太到相關資料，我也是跟朋友討論了很多次，每天大概 4~5 小時，持續快兩周，才免強完成 Project 的要求，但仍有不小的進步空間，也希望自己未來假以時日能再將這份 Project 修到最好，最後也辛苦助教了，謝謝！