

# Digital Image Processing

## Final Project

主題： Seamless Refocusing

組別： 13

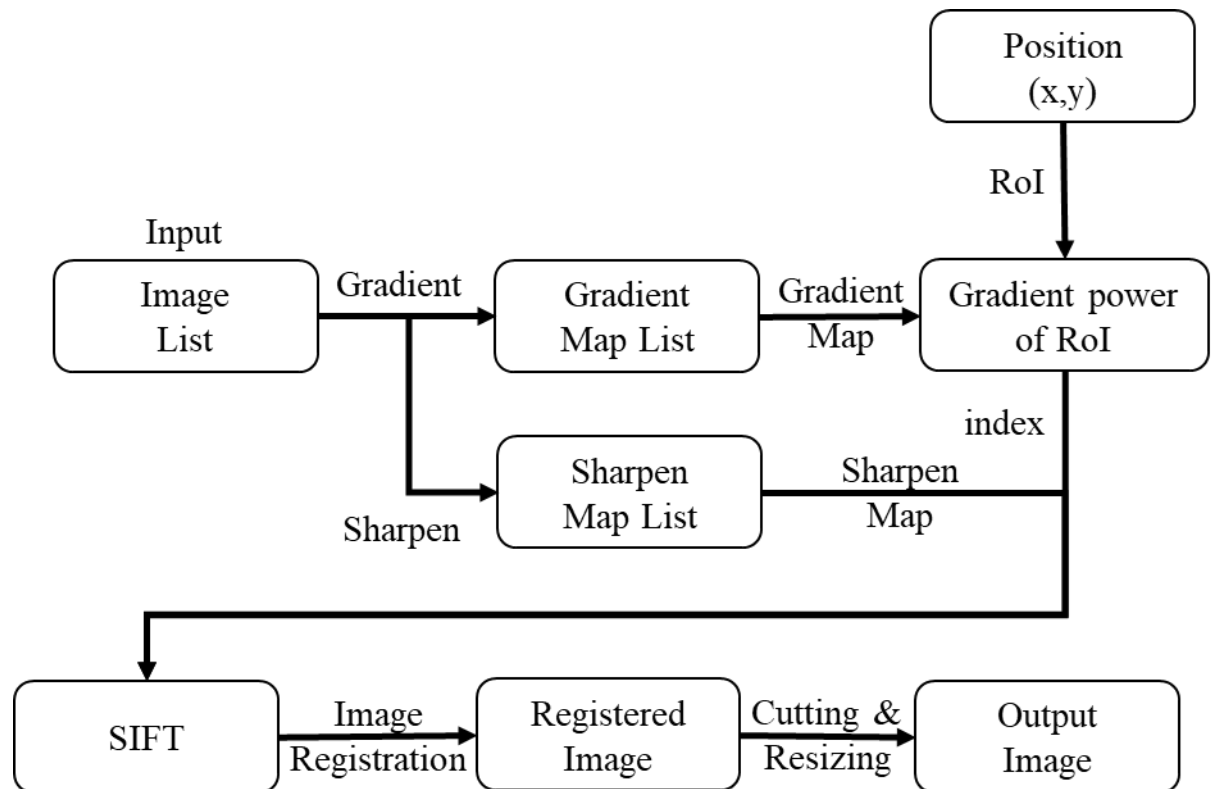
組員：

林政均 D11942011

林永平 R10943016

下圖為吾人所提出的系統架構，影像資料集輸入後，分別使用 Gradient 和 Sharpen 找出 Gradient Map 及 Sharpen Map，Gradient Map 做為使用者選取目標座標後產生 RoI，並輸出指定影像 index 的依據，Sharpen Map 則用於強化影像，以便 SIFT 的關鍵點搜索，最後使用 Image Registration 進行配準並裁切校正，最終輸出在一圖形介面上。

以下將一一解釋所使用的工具。



## 一、 Gradient (Laplacian)

1. 目的：偵測影像中的邊緣以方便做配對
2. 原理：我們利用二次微分的 Laplacian kernel(下圖右)做 convolution 來運算物體邊緣，

0	-1	0
-1	4	-1
0	-1	0

一階 Laplacian kernel

1	1	1
1	-8	1
1	1	1

二階 Laplacian kernel

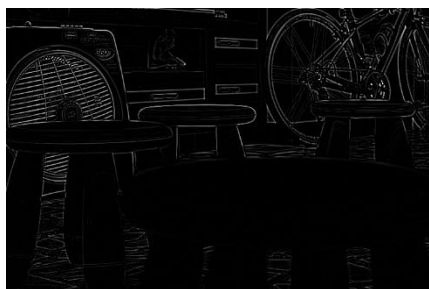
要作二次微分的原因，以數學上來說，是因為在第一次微分後，在物體邊緣處會有波峰產生，此時需要設定門檻值，如果該波峰超過門檻值，則判斷為邊緣，但是如果再經過一次微分，則微分後的波形會有通過零點(Zero Crossing)的現象，更有利於判斷物體邊緣。

3. 程式實現：

```
def gradient (img):  
    img_gray = cv2.cvtColor(img , cv2.COLOR_BGR2GRAY)  
    blur = cv2.GaussianBlur(img_gray,(3,3),1)  
    Laplacian_result = cv2.filter2D(blur, ddepth = -1, kernel = Laplacian)  
  
    return Laplacian_result
```

首先利用 cv2 的 cvtColor 將圖片轉換成灰階照片，然後再用 GaussianBlur 將灰階圖片模糊(kernel=(3,3), sigma=1)，最後套用以上 kernel 做 convolution 來輸出 edge detection 要用的照片。

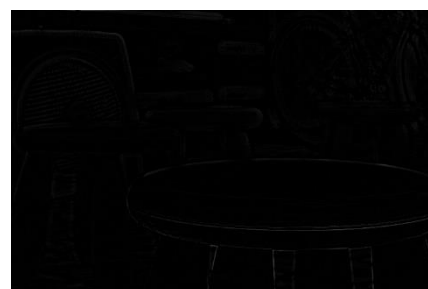
4. 結果：



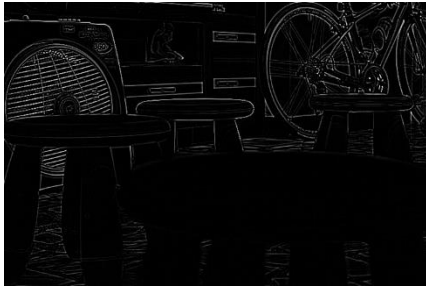
gradient\_result\_1



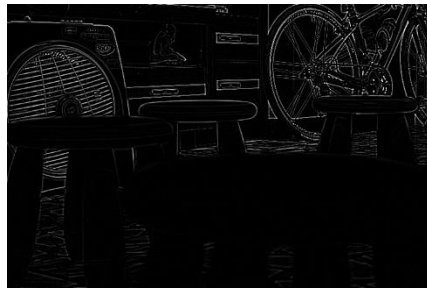
gradient\_result\_2



gradient\_result\_3



gradient\_result\_4



gradient\_result\_5

## 二、Sharpen

1. 目的：銳化可以增強影像特徵使影像在做 SIFT 時能夠更精準配對
2. 原理：利用以下 Kernel 對原圖逕行 convolution 即可銳化影像

-1	-1	-1
-1	9	-1
-1	-1	-1

3. 程式實現：

```
def sharpen (img):  
    blur = cv2.GaussianBlur(img,(3,3),1)  
    Sharpen_result = cv2.filter2D(blur, ddepth = -1, kernel = Sharpen)  
  
    return Sharpen_result
```

首先利用 cv2 的 GaussianBlur 將圖片模糊(kernel=(3,3), sigma=1)，最後套用以上 kernel 做 convolution 來輸出影像銳化後的照片。

4. 結果：



sharpen\_result\_1



sharpen\_result\_2



sharpen\_result\_3



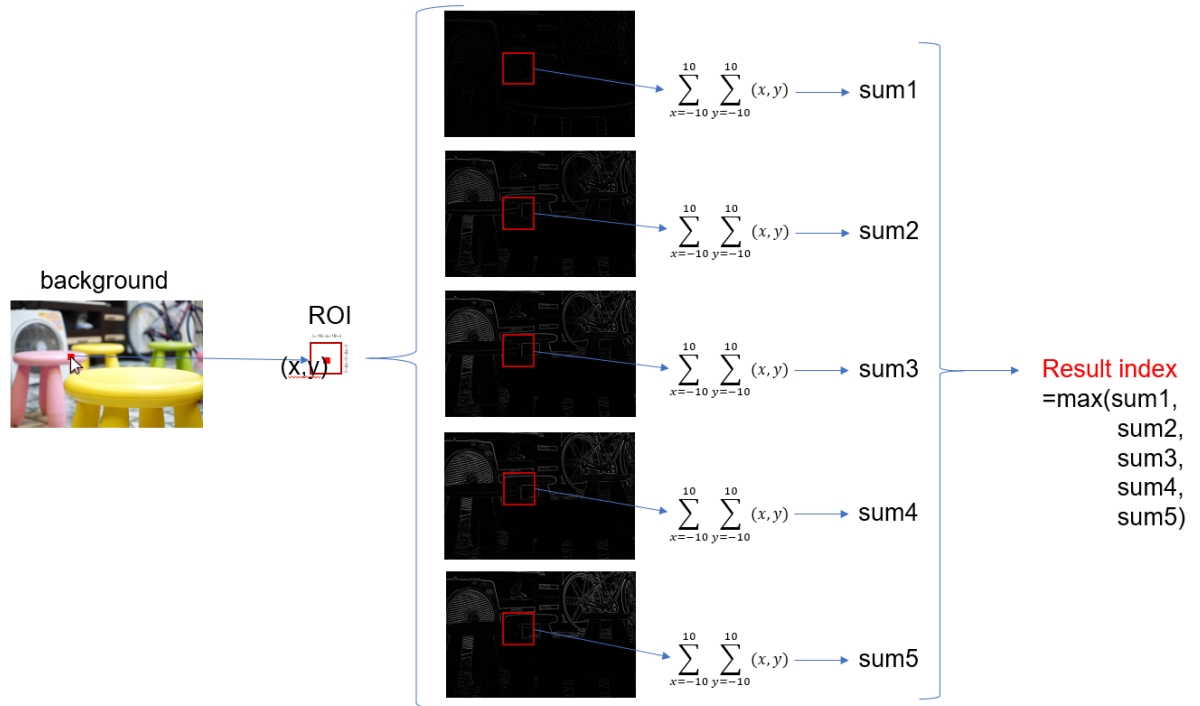
sharpen\_result\_4



sharpen\_result\_5

### 三、RoI (Region of Interesting) set

說明：讀取點擊的座標  $(x,y)$ ，RoI 範圍設為  $((x,y)-10,(x,y)+10)$  之間，並將 RoI 內的梯度圖取出並做加總，越清晰的影像，細節越清楚，梯度總和就越高。



## 四、尺度不變特徵轉換

尺度不變特徵轉換 (Scale Invariant Feature Transform, SIFT) 是一種搜索局部特徵的演算法，是 1999 年由 Lowe 提出的，其演算法分成四個步驟：

- (1) 尺度空間極值檢測
- (2) 關鍵點定位
- (3) 關鍵點方向
- (4) 關鍵點描述

### 1. 尺度空間極值檢測

特徵點在影像中是帶有較大資訊量的點，其不易受光照、雜訊、線性轉換等影響，若任意兩張圖當中，皆具有相同的物件，其特徵點位置亦相同，僅因攝影時位置不同，而產生尺度、方向、大小等差別，而文獻[]提到，高斯核是唯一可以產生多尺度空間的核函數，假設一原始影像為  $I(x,y)$ ，則該影像之尺度空間  $L(x,y,\sigma)$  可用二維高斯卷積表示，即

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

其中

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

經過一系列的高斯卷積後，影像愈加模糊，但由於特徵點不易受干擾的性質，且高斯核中心點權重為最大值，鄰近像素值較權重較小，進行多次加權平均後，相鄰的像素皆與目標像素差異逐漸增大，使得特徵點得以被保留

### (1) 高斯金字塔

由於二維影像中，同時存在重要與不重要的資訊，因此吾人可利用降採樣 (Down-sampling) 的方式，逐步地將尺度進行縮減，而後針對每一尺度進行高斯模糊，如此建立出各尺度的特徵圖 (Feature Map)，由大到小，由下至上，宛如金字塔樣式，即高斯金字塔。為求降採樣後，各尺度之間可更加連續，因此在降採樣後可利用高斯濾波進行簡易的平滑化 (smooth)

### (2) 高斯差分金字塔

在進行特徵檢測時，諸如邊緣、角點等等，通常會利用微分找出邊緣或角點等特徵點，而 SIFT 演算法中，高斯一拉普拉斯運算子 (Laplacian of Gaussian, LoG)，可使用高斯差分法 (Difference of Gaussian) 近似，即

$$\frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k(\sigma - 1)}$$

其中  $k$  為常數，並不影響極值位置，因此在實務上，僅需將每一層的上下兩層相減，即可得到高斯差分圖

## 2. 關鍵點定位

### (1) 定位

由於數位影像係由離散的形式表現，求得的極值未必為真實極值，為確保吾人所篩選出的關鍵點皆為正確，首先利用尺度空間的泰勒展開式 (Taylor Expansion)，進行擬合，其擬合函數為

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X, X = (x, y, \sigma)$$

其中  $X$  可表示為與插值中心的偏移量，當  $X$  大於某一個吾人所設定的閾值 (Threshold)，意即該插直中心已偏離甚遠，此時需改變目前的關鍵點位置，並在新的位置進行擬合，如此反覆直到收斂為止。

### (2) 去除不穩定極值

影像的邊緣亦會有極值出現，因此需要將其去除，由於圖像中的物體的邊緣位置的點的主曲率一般會比較高，因此我們可以通過主曲率來判斷該點是否在物體的邊緣位置。某像素點位置處的主曲率可以由二維的 Hessian 矩陣  $H$  計算得到

$$H = \begin{bmatrix} D_{xx}(x, y) & D_{xy}(x, y) \\ D_{yx}(x, y) & D_{yy}(x, y) \end{bmatrix}$$

若此矩陣有兩個特徵值  $\alpha$  及  $\beta$ ， $Tr(H)$  為  $H$  的對角元素和， $Det(H)$  為行列式值，則

$$Tr(H) = \alpha + \beta$$

$$Det(H) = \alpha\beta$$

首先去除行列式為負的點，而後去除主曲率較大的點，可用以下規則去除

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\gamma\beta + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma}$$

$\gamma$  越大，則該點有越大的機率位在影像邊緣上，因此需設置一閾值  $\gamma_0$ ，即

$$\frac{Tr(H)^2}{Det(H)} < \frac{(\gamma_0 + 1)^2}{\gamma_0}$$



### 3. 關鍵點方向

為實現關鍵點的旋轉不變性，尚須納入關鍵點的角度，可由高斯尺度途中得出，若高斯尺度為  $p$ ，則

$$r = 3 \times 1.5p$$

而梯度振幅為

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

梯度相位為

$$\theta(x,y) = \tan^{-1} \frac{(L(x,y+1) - L(x,y-1))}{(L(x+1,y) - L(x-1,y))}$$

最後利用直方圖對所有關鍵點做相位記錄，數位影像中則可使用角度離散化的方式分割成多個方向，建立好後再對直方圖  $h(i)$  進行平滑化，最終得到  $H(i)$

$$H(i) = \frac{h(i-2) + h(i+2)}{16} + \frac{4 \times h((i-1) + h(i+1))}{16} + \frac{6 \times h(i)}{16}$$

### 4. 關鍵點描述

經由上述步驟，吾人已得到關鍵點及其位置、相位，將關鍵點附近的區域劃分為  $d \times d$  個子區域，每一個子區域由 1 個種子及 8 個方向組成，為保證關鍵點具方向不變性，因此需將關鍵點及週遭區域進行旋轉，旋轉角度即為關鍵點相位，為使整個子區域旋轉，則旋轉半徑應為此子區域正方形對角線的一半，即

$$r = \frac{3\sigma(d+1)\sqrt{2}}{2}$$

像素點旋轉則使用旋轉矩陣進行，最後再使用三線性插值 (Trilinear Interpolation)，將像素點依序填入直方圖中。

但由於模糊影像本身資訊不足以使 SIFT 找到足量的關鍵點，因此吾人先使用前述的銳化進行影像強化，再使用 SIFT 進行關鍵點搜索，最終結果如下圖所示



## 五、影像配準

有了關鍵點匹配，吾人即可利用影像配準 (Image Registration) 的方法，將兩張影像配準在相同相位及尺度上，這裡選用 K-最鄰近演算法(K Nearest Neighbor, KNN) 做為分類演算法，如下圖所示。

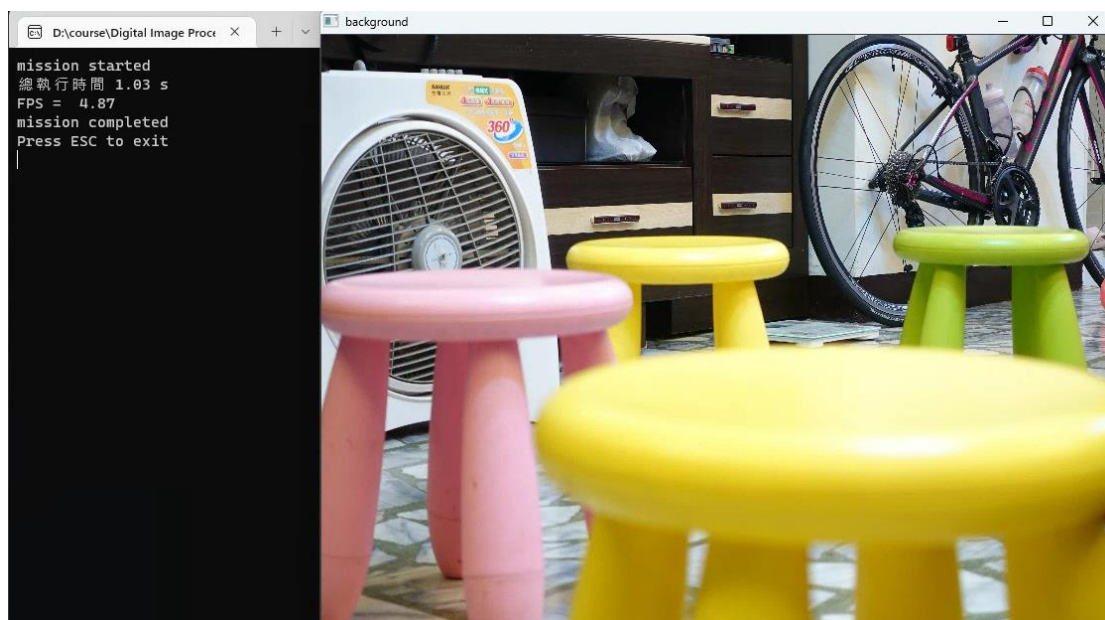


最後將超出背景的部分切除，再將尺寸放大回原本的尺寸，即配準完成，其結果如下圖所示。



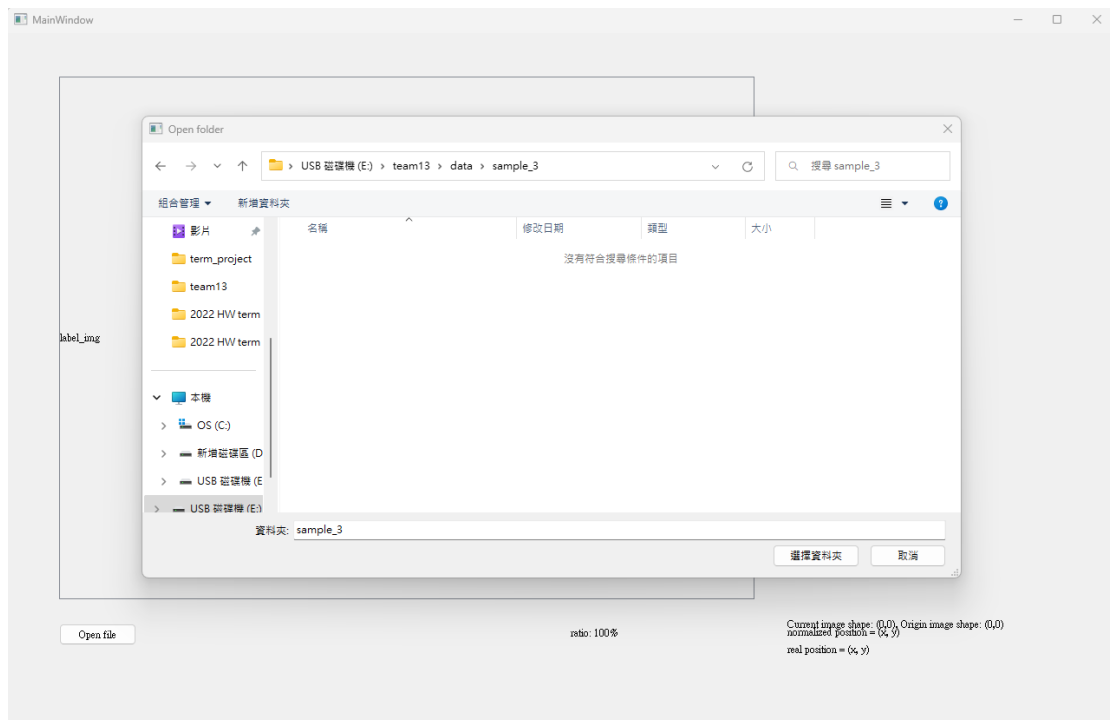
## 六、結果

吾人所提出的方法，其演算法運行效能接近 5 FPS，本應使用 SURF 進行測試，但因專利問題，致使 OpenCV 函式庫中的 SIFT 與 SURF 混亂，因此僅以 SIFT 做測試，未來若採用 SURF 或更多 SOTA 演算法，應有望可達到 real-time。

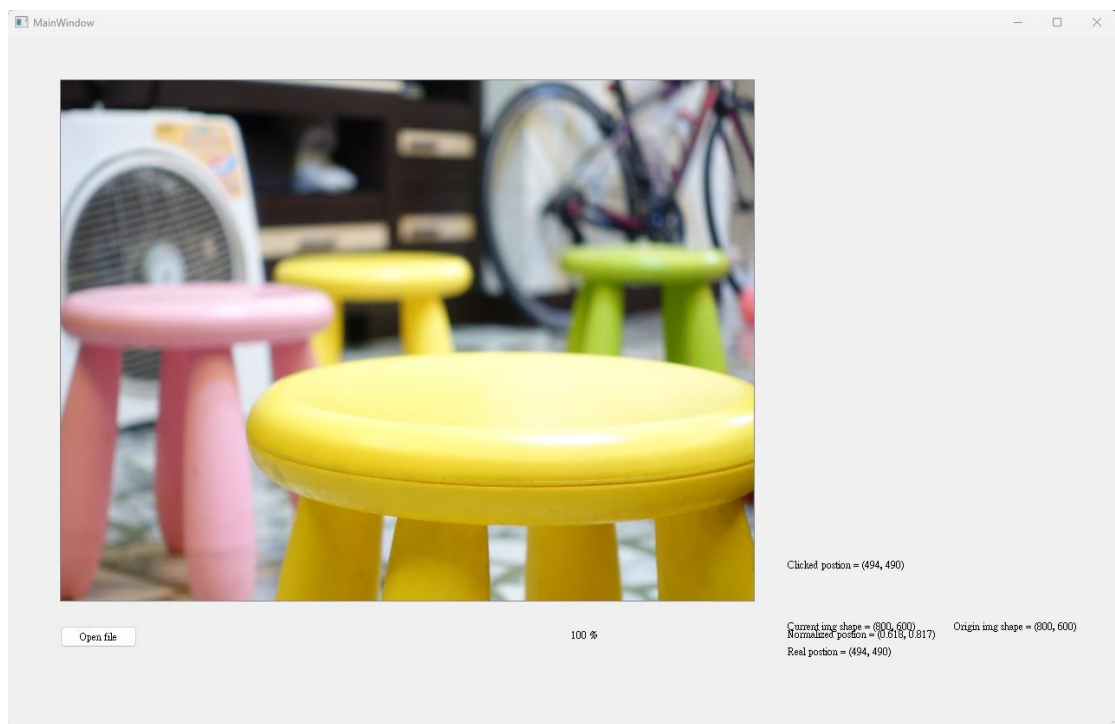
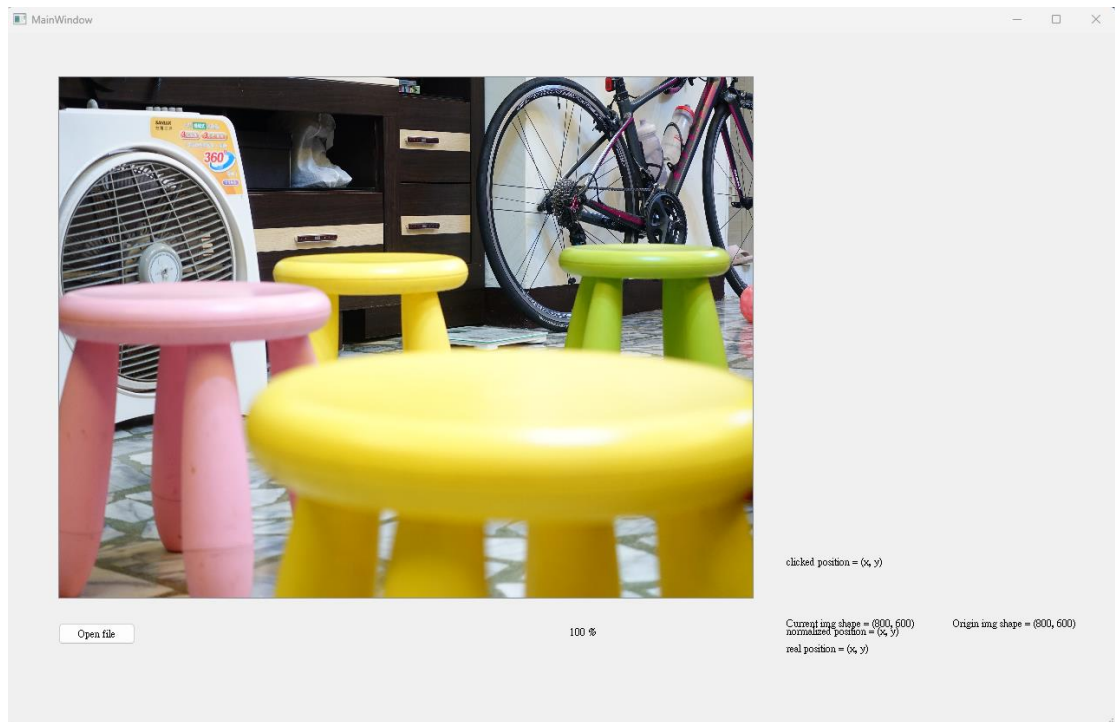


## 七、圖形介面

為方便使用者使用，吾人利用 PyQt5 設計一圖形介面 (Graphical User Interface, GUI)，使用者只要將事先拍攝好的影像放入相同路徑，再點選 Open file 找到該路徑，即可直接輸出結果，在圖形介面上點選即可切換至不同焦距的影像，使用流程如下四張圖所示。






















## 附錄、Camera Difference:

camera	type	Zoom in/out of far focal point
iPhone	cellphone	Zoom out
RX100M5	camera	Zoom in
A6600+sel1655G	MILC	unchanged

camera	Example Nearest————→farthest				
iPhone					
RX100M5					
A6600+sel1655G					

說明：我們發現不同的相機在做 focal-stack 時會有不一樣的 zoom-in/out 現象，如上表 iPhone 會隨著 focal point 越遠而 zoom out；類單 RX100M5 會有相反的現象，會隨著 focal point 越遠而 zoom in；而單眼 A6600+sel1655G 則完全不會有 zoom-in/out 現象，推測是 sel1655G 內部的對焦鏡片結構可以自行修正此現象，但因為體積限制，僅有單眼鏡頭能夠做到這樣的修正，類單或手機則無法修正，因此需藉由演算法才能夠實現無縫對焦。也因為不同鏡頭會有不同 zoom-in/out 現象，因此我們在最後一個階段做完 SIFT 和 registration 之後會判斷哪張圖是最小的，並以此圖當作新的 background，而要輸出的照片就會以這個新的 background 為基準進行 crop，此方法可以有效解決在不同裝置下做 focal-stack 時的 zoom-in/out 現象。