

Advanced Computer Vision

HOMEWORK 3

Lab : VPILab

Advisor : Cheng-Ming Huang(黃正民)

Student : Yu Cho(卓諭)

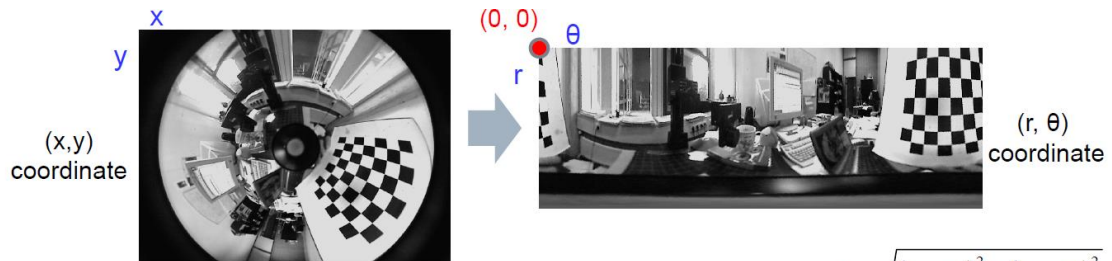
Student ID : 106318025

Data : 2017.11.30

1. Rectify the image of omnidirectional camera to a panoramic image

(a) Problem

Rectify the image of omnidirectional camera to a panoramic image



■ Input image size: 1024*768

□ $(x_c, y_c) = (512, 384)$

■ Output image size: 720*384

□ 1 pixel of $\theta = 2\pi/720$ rad.

forward warping
(x, y) to (r, θ)

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$
$$\theta = \tan^{-1} \left(\frac{y - y_c}{x - x_c} \right)$$

inverse warping
(r, θ) to (x, y)

$$x = r \cos \theta + x_c$$
$$y = r \sin \theta + y_c$$

(b) Method

使用 invers warping，依據要輸出影像的各個座標位置，經過下面公式對應回原影像座標。

$$x = r \cos \theta + x_c$$

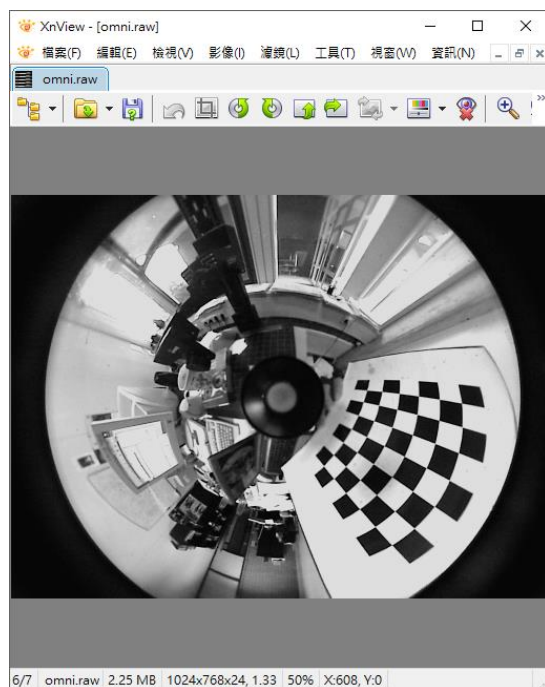
$$y = r \sin \theta + y_c$$

```
double t = ((double)theta / 720) * 2 * M_PI;  
int x = (r*cos(-t) + 512);  
int y = (r*sin(-t) + 384);
```

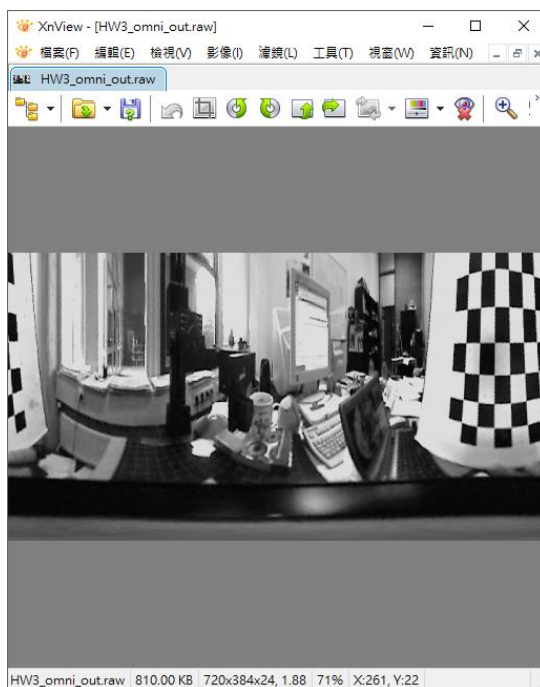
使用三個 for 迴圈分別對列、行以及通道逐行逐列掃描，並將(x, y)坐標系對應到的原影像像素點，存入(r, θ)坐標系的輸出影像。

(c) **Result**

(Original)



(Output)



(d) **Discussion**

在轉換的過程中，要注意的是，因為全向影像是 360° 的環景，而要拉伸至 720 個像素點，必須做正規化，1 pixel of $\theta = 2\pi/720$ 。將原影像存入輸出影像的步驟中，必須讓像素點由輸出圖左下開始存入，否則影像會上下顛倒(*.raw 是從左上存入像素值)。因為 r 是從 0 開始遞加，相當於從原影像的中心點往外推，中心點應在空間中的底部位置，越從中心點外推至圓邊界則越靠近空間中的天花板位置。

2. Image stitching with the projective transform

(a) Problem

Image stitching with the projective transform

■ Refer to "Projective mappings for image warping, pdf"

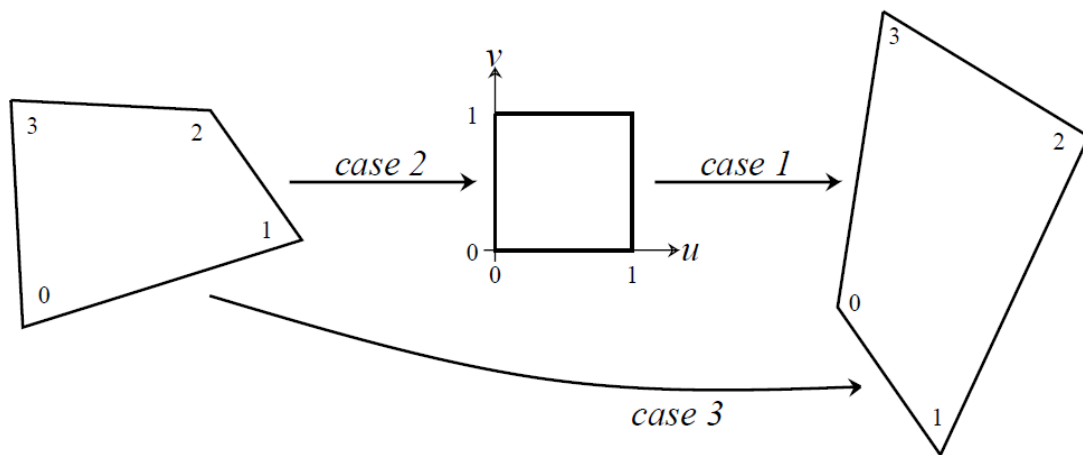
- Create an image with size 900*480 (0,0) v
- Paste the left image at [80:449, 0:509] u
- Evaluate the transformation matrix M by Eq.(3) or Eq.(4)
 - Set $(x_0, y_0) = (130, 250)$, $(x_1, y_1) = (470, 310)$,
 $(x_2, y_2) = (475, 900)$, $(x_3, y_3) = (0, 770)$
- Use inverse warping ($P_s = P_d M_{ds}$) to transfer the original (u, v) coordinate to the desired (x, y) coordinate
 - Hint: $i=1$, $w=1$, $(u, v) = (u'/q, v'/q)$



(b) Method

Input image: left.raw (510*370) 、right.raw (510*370)

Output image: HW3_image_warping.raw (700*480)



Step1. 將 right. raw 經過上圖的 Case2 轉至單位圖像，單位圖像位置

$$\text{為}(u, v) = \left(\frac{x}{510}, \frac{y}{370} \right)$$

Step2. 要經由轉換矩陣由(x, y)計算(u, v)映射像素點

$$\begin{aligned} \mathbf{p_s} &= \mathbf{p_d M_{ds}} \\ &= (u' \quad v' \quad q) = (x' \quad y' \quad w) \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix} \\ &= (x' \quad y' \quad w) \begin{pmatrix} ei - fh & fg - di & dh - eg \\ ch - bi & ai - cg & bg - ah \\ bf - ce & cd - af & ae - bd \end{pmatrix} \\ u &= \frac{Ax + By + C}{Gx + Hy + I}, \quad v = \frac{Dx + Ey + F}{Gx + Hy + I} \end{aligned}$$

Step3. 然而生成 A,B,C,D,E,F,G,H,I 的參數由以下公式求得。

先定義，

$$\begin{aligned} \Delta x_1 &= x_1 - x_2 & \Delta x_2 &= x_3 - x_2 & \Sigma x &= x_0 - x_1 + x_2 - x_3 \\ \Delta y_1 &= y_1 - y_2 & \Delta y_2 &= y_3 - y_2 & \Sigma y &= y_0 - y_1 + y_2 - y_3 \end{aligned}$$

求參數 a, b, c, d, e, f, g, h，

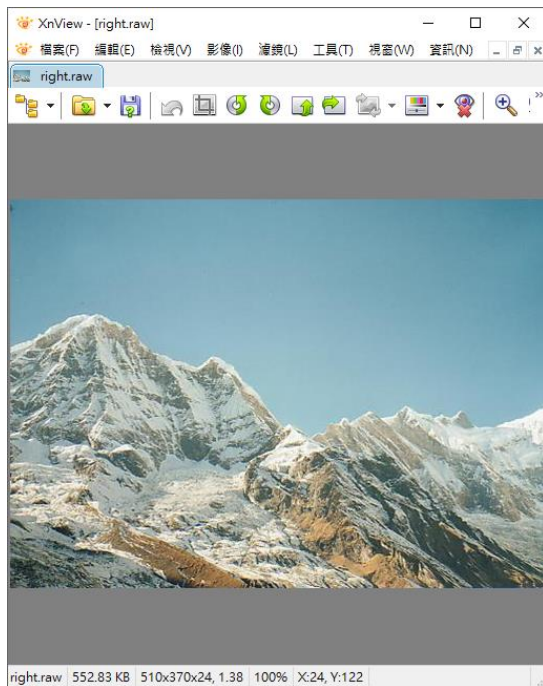
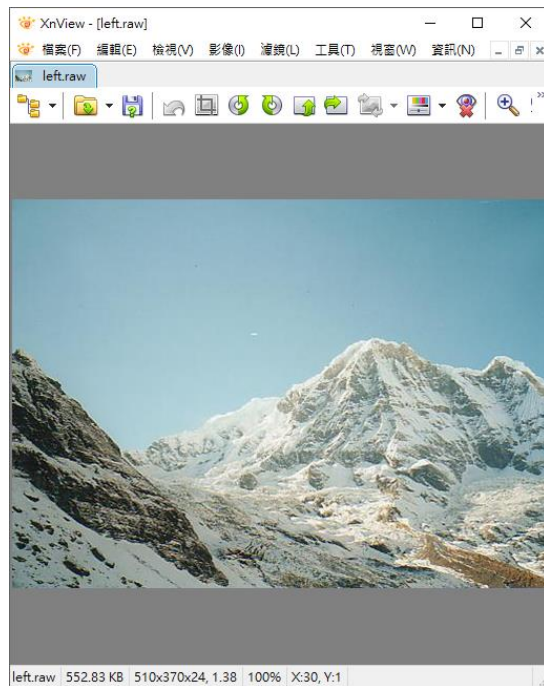
$$\begin{aligned} g &= \frac{\begin{vmatrix} \Sigma x & \Delta x_2 \\ \Sigma y & \Delta y_2 \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} \\ h &= \frac{\begin{vmatrix} \Delta x_1 & \Sigma x \\ \Delta y_1 & \Sigma y \end{vmatrix}}{\begin{vmatrix} \Delta x_1 & \Delta x_2 \\ \Delta y_1 & \Delta y_2 \end{vmatrix}} \\ a &= x_1 - x_0 + gx_1 \\ b &= x_3 - x_0 + hx_3 \\ c &= x_0 \\ d &= y_1 - y_0 + gy_1 \\ e &= y_3 - y_0 + hy_3 \\ f &= y_0 \end{aligned}$$

Step4. 將題目提供的 warping 後 4 個頂點帶入上述公式求得參數。

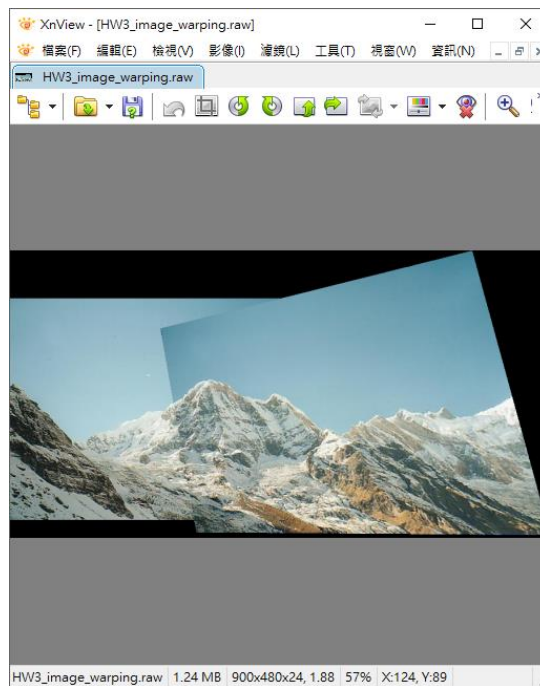
(c) **Result**

(left.raw)

(right.raw)



(HW3_image_warping.raw)



(d) Discussion

參考 Projective Mappings for Image Warping, Paul Heckbert 的論文，理解轉換矩陣的實現原理，算出轉換矩陣後可以知道要形變的影像 4 個頂點位置，並且將影像拉伸，最後將兩張影像疊合，完成 stitching，這部分只實現形變，並沒有整套的影像拼接流程，影像拼接的大致上流程為，特徵點的偵測、提取、比對，再來式影像的形變，最後做色調的一致化。若要實現影像拼接，使用 OpenCV 函式庫會較方便也更有效率實現影像拼接。

3. Bonus

(a) Problem

- Image stitching by OpenCV
 - ☐ Extract feature points
 - ☐ Find corresponding pairs
 - ☐ Compute transformations
 - ☐ Warp image
 - ☐ Blend color within overlap



(b) Method

Input : left.bmp 、 right.bmp

Output : keypoints_left.bmp 、 keypoints_right.bmp 、

Corresponding_Pairs&Warping.bmp 、 Image Stitching.bmp

Step1. 利用 SURF 提取特徵點

Step2. 特徵點描述，為之後的特徵點匹配做準備

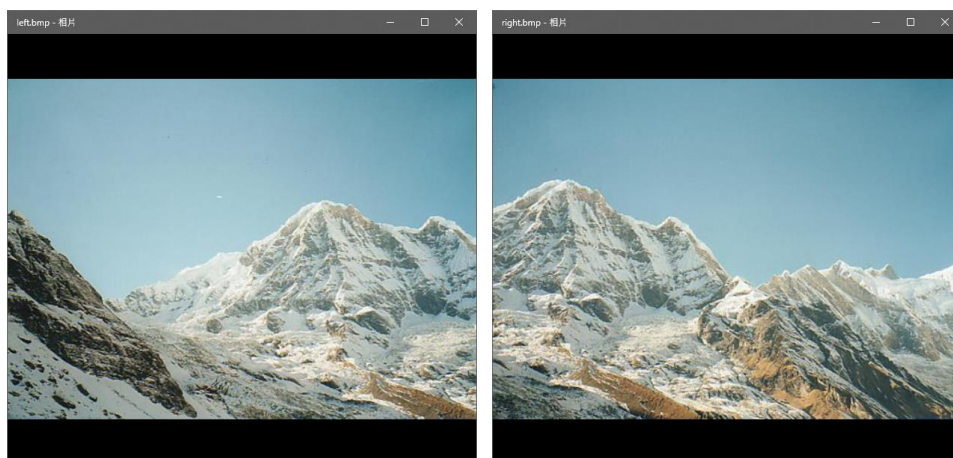
Step3. 獲得匹配特徵點，並提取最優配對(定義匹配距離小於 3 倍的最小匹配距離)

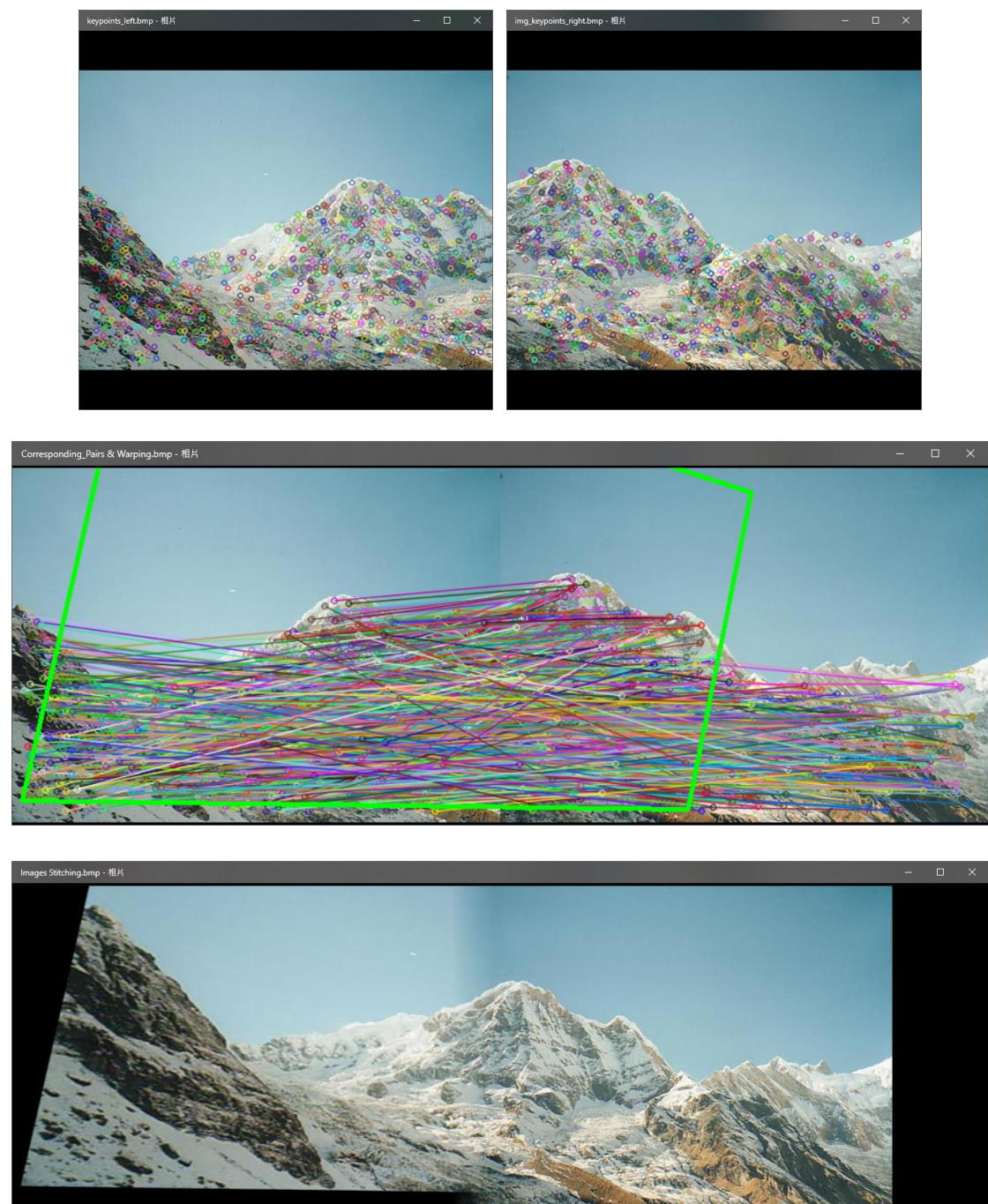
Step4. 獲取兩影像之間的投影映射矩陣，尺寸為 3×3

Step5. 獲取最佳配對點在原始圖像和矩陣變換後圖像上的對應位置，用於圖像拼接的定位

Step6. 進行接縫處色調接合

(c) Result





(d) Discussion

使用 OpenCV 函示庫實現影像拼接可以更直覺地完成每個步驟的目的，最花時間的是去理解要如何使用 OpenCV 函示庫去實現各個步驟，相關的函式使用方式都可以從 OpenCV 官網上找到。