

黃政維

R09521603

May 5, 2021

Homework 2

Please place the data in the same directory with code

Q1 Code Usage

python3 2d3dmathcing.py [version]

set version to 1 to use opencv solve pnp

set version to 2 to use mine p3pRansac solver

Q1-1

利用上課講義所寫的P3P搭配Ransac來求出每張相片得旋轉以及平移。特徵點單純利用助教程式碼的方法尋找匹配點。

Ransac Parameter

Ransac 參數部分我沒有特別調參過，根據講義上的符號定義我的參數如下

s : 4

e : 0.5

p : 0.99

d : 10

由上述公式計算出需要 sample 的次數，因為只有大約七十上下，因此沒有做 early termination，計算 inlier 方式為將點雲投影到相片上計算與真實像素的 L2-norm，這個誤差須在十以內，實驗結果發現以這樣的參數下每張相片大約有一千五至兩千五個 inlier。

Trilateration

在求得三個方向的長度後，我是利用三角定位法來求得相機位置的，一開始的想法

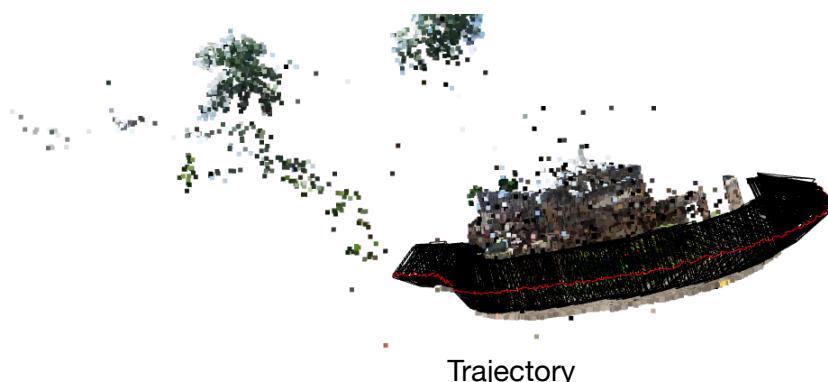
是利用距離公式寫出三條分別距離 x_1 x_2 x_3 距離的圓公式，並且兩兩相減得到三條三個未知數的方程式再利用 inversion 求解相機位置，但後來發現這個方式有問題，在求解 inversion 時都是 singular case，後來發現並不能這樣解，因此更改成正確的方式。

後來的方式則為，先求 x_1 x_2 x_3 中 x_1 與 x_2 的向量，作為 x 軸，z 軸則為 x_1 x_2 x_3 所在平面的法向量，將 x 軸與 z 軸外積即可得到 y 軸，再利用三角形基本定理求解相機座標的位置，詳情可以看 [1]

Q1-2

畫圖部分，我將相片四個角落位置，分別是 $(0, 0)$, $(1080, 0)$, $(1080, 1920)$, $(0, 1920)$ 依據每張相片得到的旋轉平移矩陣反投影到空間中，在從相機座標畫出四條線來模擬相機空間位置，運動軌跡部分，我從 X 軸最右邊的相機位置開始尋找最近相機位置畫線，畫完後就刪掉上一個相機位置，但因為我的 P3P 會有誤差，導致有些相機會在搜尋最近點時被忽略掉，但被遺漏的相機會在最後自己連成線，導致很嚴重的大距離位移，明顯不是正常運動軌跡，因此在繪製軌跡時如果軌跡長度大於 0.8 會被我忽略掉。

以下為軌跡照片，黑色代表相機，紅色代表路徑。



Q1-3

基本上依照助教寄信方法計算中間值誤差，將自己的旋轉矩陣利用 `scipy` 的 `inv()` 算出一個向量，並且在將這個向量與 `ground truth` 相乘，最後利用 `as_rotvec`，轉成 `axis angle` 並計算該向量的 `norm`。

Translation Error : 0.0289

Rotation Error : 0.2116

因為沒有固定 `random seed` 數值可能每次有差但 Translation Error 大約在 0.03 上下，Rotation Error 大約在 0.2 上下。

討論：

在實作 P3P 前期時我發現不論怎麼做我所計算出來的相機位置皆與 `ground truth` 以及 `opencv` 所計算出來的差異很大，因此花費了滿多時間在檢查哪裡做錯，後來實在是花費太多時間因此只好先做 Q1-2，然而在實作 Q1-2 的時候才發現如果以 `ground truth` 的相機位置畫圖的話，需要將相機位置做 $T = -\text{inv}(R)@T$ 的轉換，也就是說我原本的作法並沒有錯，只是如果需要對照 `ground truth` 的話需要將我的相機位置先做旋轉，後來我在 `opencv` 的 source code 發現他們是先算出旋轉矩陣才返算相機位置的，因此才會有這個誤差，另外有實作處理 `distortion`，但結果比沒做 `distortion` 還糟一點，我認為應該是方法上有錯誤，但來不及修改，因此最後繳交的仍是沒有處理 `distortion` 的方法。

Q2 Code Usage

python3 VRvideo.py [version]

set version to 1 to use opencv solve pnp

set version to 2 to use mine p3pRansac solver

Q2-1

需要先執行 Q1 的程式碼產生旋轉平移矩陣後才可以執行 Q2 的程式。

先利用助教所提供的 code 將 cube 八個角點在三維空間座標找出來，在用向量方式在八個面上加，加點方式為在該面上選取三角點產生兩向量，並且組合兩格向量產生滿足夠的點，再利用與每張 valid 相片相機位置之距離從遠到近排序，利用第一大題所求得之旋轉平移矩陣計算點在相片上的像素，逐點繪製上去。從結果上來看可以發現如果使用我自己的 P3P 方法方塊大致上也是正確的，只是會較為晃動，並沒有那麼平穩。



Virtual box

Reference

- [1] Pablo Cotera¹ , Miguel Velazquez¹ , David Cruz¹ , Luis Medina¹ and Manuel Bandala¹. 20, March, 2016. Indoor Robot Positioning using an Enhanced Tri lateration Algorithm