

3DCV HW3 Report

電子所碩一 R10943117 陳昱仁

一. Introduction

這次實作的 Visual Odometry，使用 NTU dataset，整體分為以下幾個步驟

Pseudo code: Visual Odometry

Step 1 Camera calibration

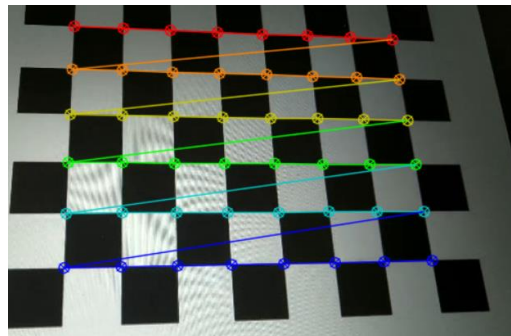
Step 2 Feature Matching

Step 3 Pose from Epipolar Geometry

Step 4 Results Visualization

二. Camera Calibration

利用助教是先給好的 camera_calibration.py，在適當時機按下空白鍵，計算 camera intrinsic matrix，結果如下圖所示。



```
Camera Intrinsic
[[522.51834977  0.          316.74694631]
 [  0.          523.86618991 182.2295255 ]
 [  0.           0.           1.          ]]
Distortion Coefficients
[[ 1.17401590e-01 -8.18681868e-01 -1.03559439e-03  1.33196131e-04
  1.47411602e+00]]
```

三. Feature Matching

由於 VO 相當注重 real time 的實現，因此在兩相鄰的 frame，我們不使用常見且 robust 的 SIFT，這裡使用 ORB 來取代 SIFT 以達到更快的速度。呼叫 `cv2.BFMatcher(cv.NORM_HAMMING, crossCheck=True)` 來做 feature matching。

四. Pose from Epipolar Geometry

使用 `cv2.findEssentialMat` 求出 Essential matrix 再用 `cv2.recoverPose` 將 E 分解出兩 frame 相對的 R 及 t。值得注意的是，這裡求出的 t 還需要一個 scale 才能還原出 pose，故需要找出對應的 3D 點，並如下圖公式，找出兩個 frame 的 t 相對的 ratio。

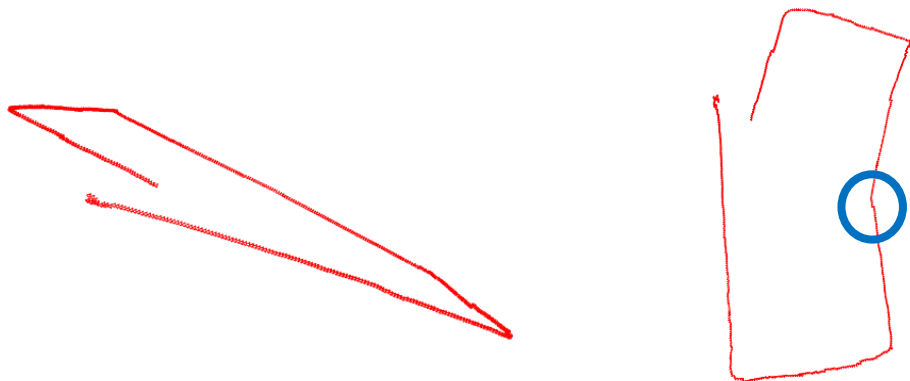
$$\frac{\|{}^{k-1}\mathbf{t}_k\|}{\|{}^k\mathbf{t}_{k+1}\|} = \frac{\|{}^k\mathbf{X}_{k-1,k} - {}^k\mathbf{X}'_{k-1,k}\|}{\|{}^k\mathbf{X}_{k,k+1} - {}^k\mathbf{X}'_{k,k+1}\|}$$

五. Results Visualization

同步顯示軌跡圖(open3d)以及每張 frame 偵測到的 key point(opencv)，並將 key point 畫在圖上。



觀察兩張軌跡圖，一張是側拍圖，一張是俯瞰圖，可以看見在最後環並沒有接在一起，應該是在藍圈處的 rotation 算的不夠準確導致。



六. 結果與討論

在這次實作中，如上方描述，藍圈處 rotation 算得不夠精確，若修正此處，應可以算出完美的環。此處的畫面如下圖，為一台白色汽車開過計中前面那條路的畫面，在實作時也會發現此處的 matching feature 相較其他圖片少很多，導致誤差產生，一方面也可視為，若 video 中有高速移動的動態物體，則實作的 VO 並不能有效的求出 pose。

