TDCV HW2 Report

r10922061 資工所 王竑睿

Problem1

Q1-1 For each validation image, compute its camera pose with respect to world coordinate. Find the 2D-3D correspondence by descriptor matching, and solve the camera pose. Implement at least one kind of algorithm that solves a PnP problem. Briefly explain your implementation and write down the pseudo code in your report.

1. 本次作業使用的是 P3P+RANSAC algorithm

```
### 取得第 idx 張 2D Image 上的特徵點 kp_query 以及他們對應的 descriptor desc_query
rimg, kp_query, desc_query = get2DImgDescriptor(idx, valid_imgs)
### 取得所有 3D 點的 descriptor (將這些3D點在所有2D images上的 descriptor平均，以作為3D點的descriptor)
kp_3Dpoints, desc_3Dpoints = get3DpointsDescriptor(train_imgs, points3Ds)
points2D, points3D = get2D3Dcorrespondence((kp_query, desc_query), (kp_3Dpoints, desc_3Dpoints))
### 使用 p3p 估測出 camera 外在參數
Random select 3 2D-3D pairs and input to p3psolver
def p3psolver(points2D, points3D, cameraMatrix):
    v = cameraMatrix.inv()* points2D # v[0], v[1], v[2] are 3-dim vectors
    ### solve the translation T of camera 外在參數
    c01, c12, c02 = cosine(v[0],v[1]), cosine(v[1],v[2]), cosine(v[0],v[2])
    r01, r12, r02 = norm2(point3D[0], point3D[1]), norm2(point3D[1], point3D[2]), norm2(point3D[0], point3D[2])
    T is the point with distance a,b,c from point3D[0], point3D[1], point3D[2]
    a,b,c could be derived from solving 4-order polynomial from c01, c12, c02, r01, r12, r02
    with distance a,b,c, T could be calculated using trilateration
    lambda could be solved with lambda* |v| = (x-T)
    Rotation matrix R could be derived from lambda*v*(x-T).inv() # x-T is invertible due to 3x3 matrix with 3 points
    return R, T
RansacSolveP3P repeat 100 iterations for each validation image
Each iteration acquire an (Ri,Ti) from 3 randomly chosen 2D-3D correspondence
The (Ri,Ti) accept other 2D-3D correspondence as an inlier when
the 3D point projected by (Ri,Ti) is no further from the 2D point than 1 pixel
return (Ri,Ti) with the most inliers from (R1,T1), (R2,T2), ..., (R100,T100)
```

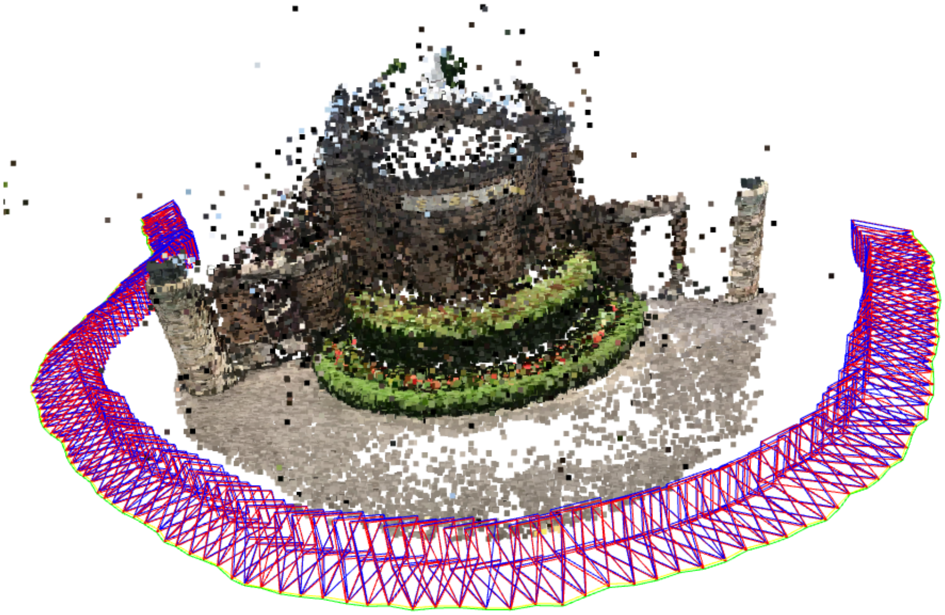
Q1-2 For each camera pose you calculated, compute the median pose error (translation,
rotation) with respect to ground truth camera pose. Provide some discussion.

1. median translation error
   a. 0.014407892836294562
2. median rotation error
   a. 0.0015140071106522573

3. 計算 median rotation error 時，是使用如下方式：
    a. 將 ground truth quaternion 以及 p3p 求出的 quaternion 化為 rotation matrix Rgt, R
    b. 計算相對旋轉 Rrel = Rgt.T* R
        i. 由於旋轉矩陣是 orthonormal 的，所以將其轉置就能取得反矩陣
    c. 再將相對旋轉矩陣 Rrel 化為 quaternion 並且依照作業投影片的公式化為 axis-angle representation，angle 即為所求。
    d. angle 愈大，表示 p3p 算出的旋轉與 groundtruth 差愈多。

Q1-3 For each camera pose you calculated, plot the trajectory and camera poses along with 3d point cloud model using Open3D. Explain how you draw and provide some discussion.

1. 一個 pinhole camera 由成像平面以及透視中心構成，故示意圖可以畫成一個金字塔，塔頂是透視中心，塔底是成像平面。
2. 由於成像平面落在 $z = 1$ 且透視中心位在原點，因此一個 camera 初始化時應該是由5個頂點所構成 (0,0,0)、(f* r, f* r, 1)、(f* r, -f* r, 1)、(-f* r, f* r, 1)、(-f* r,-f* r, 1)
    a. 其中 f 表示 focal length
    b. r 表示 aspect ratio
    c. 塔頂是 (0,0,0)，其餘4點構成塔底，形成金字塔
3. 根據每一張 image 的 camera pose (Rotation, Translation) 來將初始化 camera 移動到拍攝該張 image 時的位置與方向。
    a. 對5個 camera 的初始點 $P_i = (x, y, z), i = 1, 2, ..., 5$ 進行如下運算
        i. $\hat{P}_i = R^{-1} * (P_i - T)$
        ii. 這是因為一般將物體投影到 camera 成像平面，與將 camera 移動到拍攝位置正好是兩個互相 inverse 的操作，但兩種操作會使 camera 和 object 產生相同的相對位置。
    b. 下圖是將 130 張 validation image 的 camera pose 全部進行計算
        i. 紅色為 p3p 計算出的 camera pose，綠色為其 camera trajectory
        ii. 藍色為 ground truth camera pose，黃色為其 camera trajectory



Problem2

Q2-1 With camera intrinsic and extrinsic parameters, place a virtual cube in the the validation image sequences to create an Augmented Reality video. Draw the virtual cube as a point set with different colors on its surface. Implement a simply but efficient painter’s algorithm to determine the order of drawing.

1. 參見 virtual_cube_AR.mp4
2. 截圖如下