

3DCV&DL HW3

R11922196 林佑鑫

Briefly explain my method in each step:

Camera Calibration:

使用助教提供的 `camerat_calibration.py`，在多張圖上抓出棋盤格 corner 以及相對應的 3D corner 座標(自己定義)對其做校正，得到內部參數與形變參數。

K:

```
[[519.57919693    0.    316.83658406]
 [   0.    520.46700104 181.96843351]
 [   0.         0.         1.         ]]
```

dist:

```
[[ 1.29384673e-01 -9.44525722e-01 -1.79174961e-03  1.02536311e-03
  1.76098660e+00]]
```

之後每張照片讀取進來後都會先做 distortion，先去除鏡頭形變影響。

Feature Matching:

去形變後的影像轉成灰階，之後用 orb 抓 feature，再對相鄰兩張 image 根據 Hamming distance 做 match，得到相鄰圖片的 feature matching。

Pose from Epipolar Geometry:

迭代用 feature matching 得到的對應點算 Essential Matrix，再分解成 R, t。

用之前所有的 R, t 和目前得到的 R_k, t_k 做組合，得到相對於最開始 camera 的旋轉和平移。

根據投影片步驟對 t 做 rescale，得到用於畫圖的 t_{scale}。

1. 分別用前一次的 previous_R 和 previous_t 和這次得到的 R, t 對三角化的點算出轉換後的位置。
2. 對 previous 和 cur 算 norm，得到 scale ratio
3. 現在 t 的 norm 和 previous_t 的 norm ratio 等於 step2. 的 scale ratio，即為所求。

Result Visualization:

將圖上四個點和原點用內參的反矩陣投影回三維空間中，用這五個點表示相機的焦點和 pose 對每一個時間點，用前一步得到的 R, t 對五個點做 rotation, translation。得到他們相對於初始位置，最後用 Homework2 的 open3d lineset 方法將 pose 呈現在三維空間中。

Youtube link:

<https://youtu.be/X9Y1bv8o08s>

Please tell us how to excute your codes, including the package used and the envirment:

Hardware:

1. CPU: i7-6700k
2. GPU: Nvidia GTX 1080

Environment:

Windows 10 21H2 / Python 3.10

Packages:

1. numpy
2. opencv-python==4.5.1.48
3. open3d

環境配置是用 pip-compile 取得上面三個 package 所需的相依套件，可直接用 pip install -r requirements.txt 安裝

File structures:

```
├── calib_video.avi
├── camera_calibration.py
├── camera_parameters.npy
├── frames
├── README.md
├── report.pdf
├── requirements.in
├── requirements.txt
└── vo.py

1 directory, 8 files
```

Commands:

Environment:

pip install -r requirements.txt

Reproduce:

python3 vo.py frames