

3DCV HW2

R10922134 賴奕善

Camera calibration:

```
=> start corner detection and calibration
=> corners found in 36 images
=> Overall RMS re-projection error: 0.28923301648599564
Camera Intrinsic
[[508.78176059  0.          314.82153102]
 [  0.          509.87402467 186.94234297]
 [  0.           0.           1.          ]]
Distortion Coefficients
[[ 1.19181916e-01 -8.05201649e-01 -1.57764020e-03  2.92342337e-04
  1.38893588e+00]]
```

Feature matching:

Feature 數量:

- ORB



- SIFT



可以很清楚的看到 SIFT 找到的點遠比 ORB 來的多，會造成在運算時需要較長的時間。

計算 Process_frames 所需的時間:

- 使用 ORB: The time used to execute this is given below
181.9594259262085

- 使用 SIFT: The time used to execute this is given below
247.6088089942932

ORB 所需的計算時間比 SIFT 少，因為本身演算法就 SIFT 快，而且找的点也比較少，在後續的計算也比較快。

Pose from epipolar geometry:

找到兩個 frame 之間的 correspondences，用這些點計算 essential matrix，再把 essential matrix 分解成兩個 frame 之間的 rotation 和 translation.

```
#step2
pre_frame_points = np.array([pre_frame_kp[m.trainIdx].pt for m in matches])
cur_frame_points = np.array([cur_frame_kp[m.queryIdx].pt for m in matches])

#step3

E, mask = cv2.findEssentialMat(pre_frame_points, cur_frame_points, self.K, \
                               cv2.RANSAC, 0.999, 1.0)
#print("E",E)

#step4
retval, R, t, mask, triangulated = cv2.recoverPose(E, pre_frame_points, cur_frame_points, \
                                                    self.K, distanceThresh=1000, mask = mask)
triangulated = triangulated[:3,:] / triangulated[3,:].reshape(1,-1) # 3*N
```

Scale consistency:

從 ${}^{k-1}_k \text{Correspondences}$ 和 ${}^k_{k+1} \text{Correspondences}$ 找同樣的點 $O(n^2)$ ，隨機取兩個計算比例，並重複做 10 次，然後取中位數，當作 scale factor。取中位數是因為算出來的結果差距太大，平均值不具代表性。

Discussion:

從結果圖可以發現 SIFT 算出來的 camera pose 應該是比較準，因為有較多的 correspondences，但相對的運算時間也比較長，因此在要求 real time 時，使用 ORB 選取特徵點是比較好的選擇。

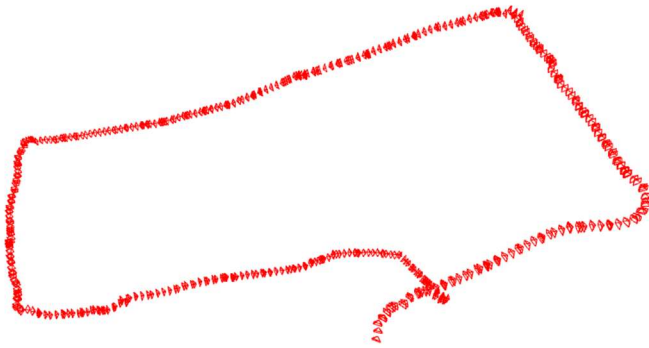
在 feature matching 時，可以保留上一張圖片找到的 keypoints 和 descriptor，讓每個 iteration 只需要做一次 detectAndCompute，減少計算量。

Usage:

```
python .\vo.py --camera_parameters .\calibrated.npy frames
```

Result:

ORB:



SIFT:

