Interdisciplinary Division of Aeronautical and Aviation Engineering
航空工程跨領域學部

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Tutorial 2: Tutorial on Visual Positioning

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Research Assistant Professor

Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 9, 16 March 2022

Interdisciplinary Division of Aeronautical and Aviation Engineering
航空工程跨領域學部

The Hong Kong Polytechnic University
香港理工大學

Preview on: Case Study Presentation

AAE4203 – Guidance and Navigation

Dr Weisong Wen
Research Assistant Professor
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University

Week 12, 16 March 2022

Requirements

> 2 or 3 students in a group
> 25 minutes for presentation and 5 minutes for Q&A
> Present your topic with PPT
> All the member in a group should present
> The presentation is suggested to including *background*, *motivation*, *methodology* (if have) and *conclusion*.

Suggested Topics

Zhang, Ji, and Sanjiv Singh. "Low-drift and real-time lidar odometry and mapping." Autonomous Robots 41, no. 2 (2017): 401-416.	**Qin, Tong, Peiliang Li, and Shaojie Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator." IEEE Transactions on Robotics 34.4 (2018): 1004-1020.**
Wen, Weisong, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter." NAVIGATION, Journal of the Institute of Navigation 68, no. 2 (2021): 315-331.	Campos, Carlos, Richard Elvira, Juan J. Gómez Rodríguez, José MM Montiel, and Juan D. Tardós. "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam." IEEE Transactions on Robotics 37, no. 6 (2021): 1874-1890.
Wen, W. and Hsu, L.T., 2021, May. Towards robust GNSS positioning and Real-time kinematic using factor graph optimization. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5884-5890). IEEE.	Engel, Jakob, Vladlen Koltun, and Daniel Cremers. "Direct sparse odometry." IEEE transactions on pattern analysis and machine intelligence 40, no. 3 (2017): 611-625.
Navigation Technique of Tesla’s Autonomous Driving	Navigation Technique of Google’s (now Waymo) Autonomous Driving
GNSS Real-time Kinematic Positioning for Autonomous Driving	Ding, W., Hou, S., Gao, H., Wan, G. and Song, S., 2020, May. Lidar inertial odometry aided robust lidar localization system in changing city scenes. In 2020 IEEE International Conference on Robotics and Automation (ICRA) (pp. 4322-4328). IEEE.

https://docs.google.com/document/d/1RujUcD1sNyUs31-2Bucn20h99OA8EtTBhY2J-XJfeDw/edit

Outline

1. Objective 1: Image calibration
   1. Zhang Zhengyou Calibration
   2. Try different number of images on chess board (**8, 16, 24**)
2. Objective 2: Estimate the visual odometry based on multiple consecutive images
   1. Structure from Motion
   2. Visual Odometry

Arrangement for the tutorial on 16rd March

- Remote access to MATLAB 2020a of QT004.
- Lecturer and Teaching Assistant will assist the tutorial online.
- using the "Win10 (Reserved)" pool of computers

Model of the camera

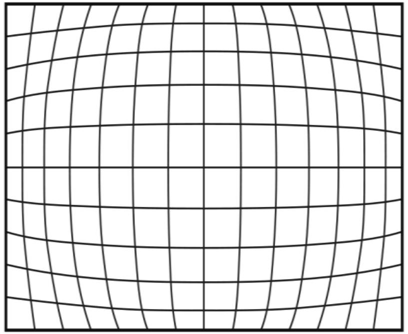
Distortion Reasons

Different degree of distortion

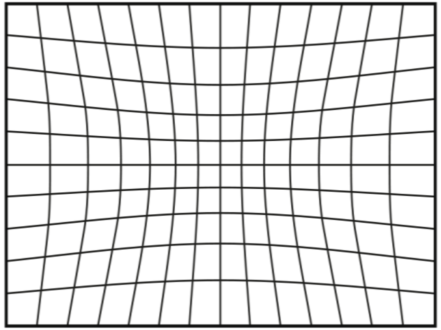
Why distortion occur?

- ➢ Optical distortion
- ➢ Assembly of the camera

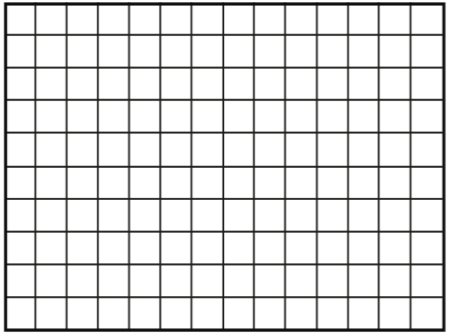
Due to **lens shape,**
called **radial distortion**



Barrel distortion
Severe distortion in the middle



Pincushion distortion
Severe distortion in the boundary



No distortion

Model of the camera

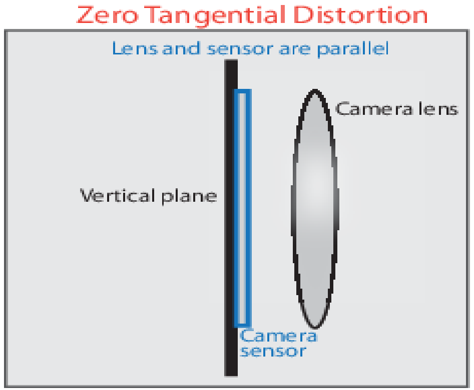
Distortion Reasons

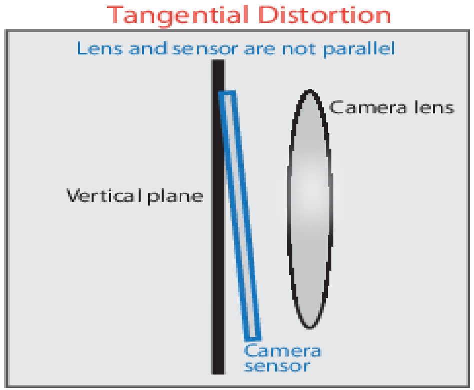
Why distortion occur?

- Optical distortion
- Assembly of the camera

Due to the **assembly error,** the lens and the imaging plane **cannot strictly parallel,** called **tangential distortion**

Different degree of distortion





Model of the camera

How to formulate these distortion ?

- Optical distortion: **radial distortion**
- Assembly of the camera: **tangential distortion**

Corrected coordinates

$$x_c = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_c = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy$$

k1, k2, and k3 — Radial distortion coefficients of the lens
p1 and p2 — Tangential distortion coefficients of the lens
r^2: $x^2 + y^2$

Illustration


Origonal Photo


Corrected photo

Model of the camera

Calibration correction

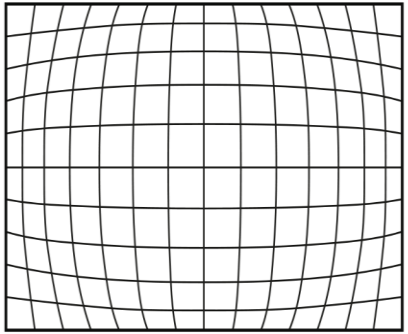
$$x_c = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_c = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y^2) + 2p_2 xy$$

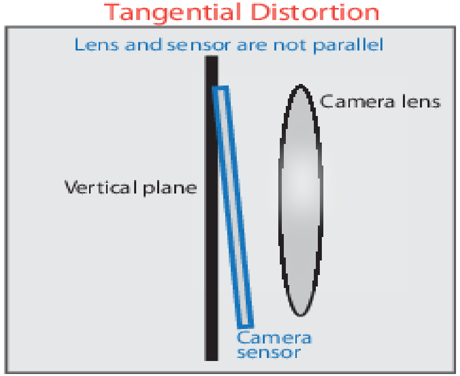
The calibration is to get **the coefficients of distortion**

k1, **k2**, and **k3** — Radial distortion coefficients of the lens
p1 and **p2** — Tangential distortion coefficients of the lens

How to calibrate ? And how many parameters to calibrate?



k1, **k2**, **k3:** radial distortion coefficients



p1, p2 :tangential distortion coefficients

Model of the camera

Calibration of camera

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z^C} \begin{bmatrix} f_u & 0 & \Delta u \\ 0 & f_v & \Delta v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^C \\ y^C \\ z^C \end{bmatrix}$$

- Optical distortion: **radial distortion**
- Assembly of the camera: **tangential distortion**

Iterms	param1	param2	param3	param4	param5
Camera Intrinsic-K	f_u	f_v	Δu	Δv	
Lens Distortion	K1	K2	K3	p1	p2

Camera Calibration

Calibration of camera

Algorithm: Zhang Zhengyou Calibration[1]

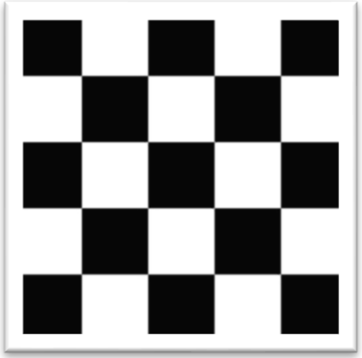
Advantages:

The equipment is simple, just a printed checkerboard;

High precision, relative error can be lower than 0.3%;

He received the IEEE Helmholtz Time Test Award for "Zhang's Calibration Method“ in 2013

A very famous expert in computer vision and multimedia technology

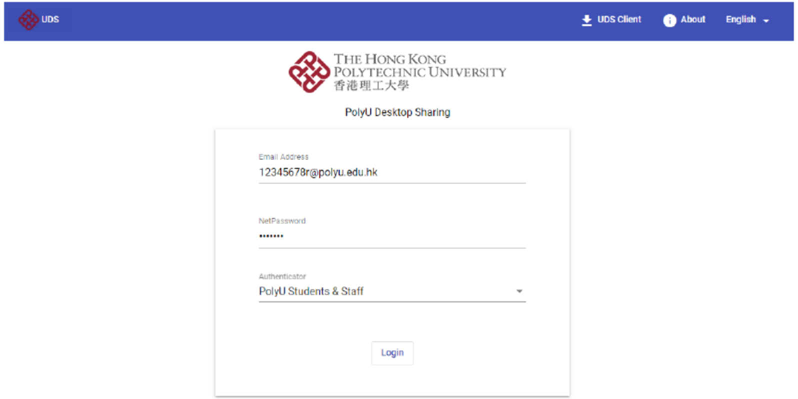


[1] Zhang, Zhengyou. "A flexible new technique for camera calibration." *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000): 1330-1334.

Arrangement for the tutorial on 23rd Feb

Remote Desktop Access Instructions

1. Remote desktop log-in address should take you to the following page. Use your full and formal PolyU Connect e-mail address to sign in.
(Address: https://puuds.polyu.edu.hk/uds/page/login)

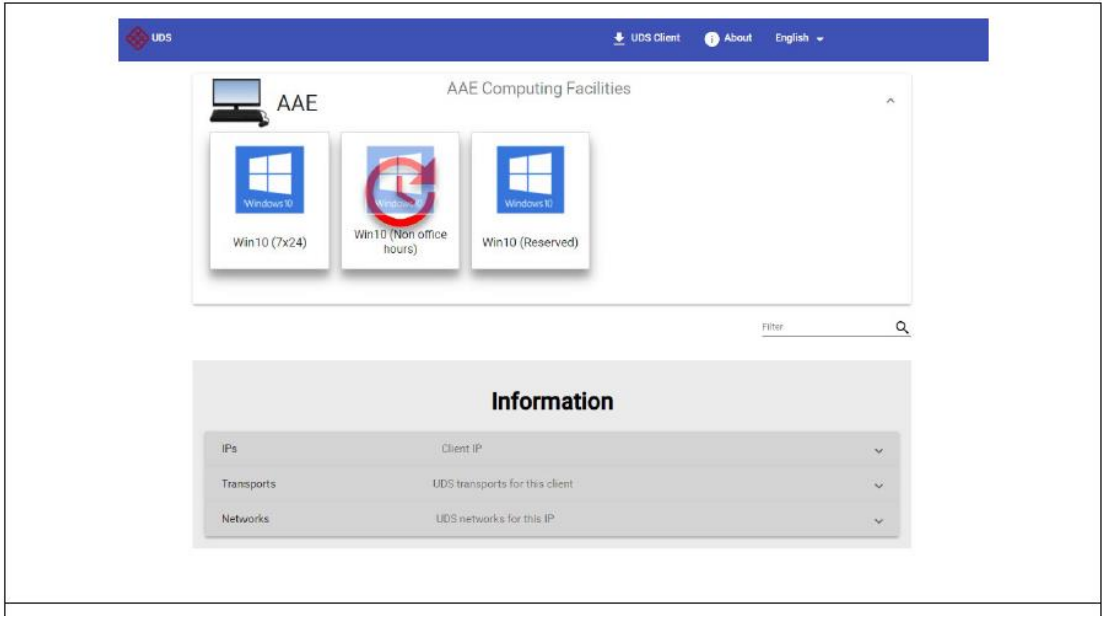


2. For general access outside of office hours: click on the “**Win10 (Non office hours)**” option. It will only be available when the lab is physically closed.
If you have reserved a PC: the “**Win10 (Reserved)**” option will be available to you.

https://puuds.polyu.edu.hk/uds/page/login

Arrangement for the tutorial on 23rd Feb

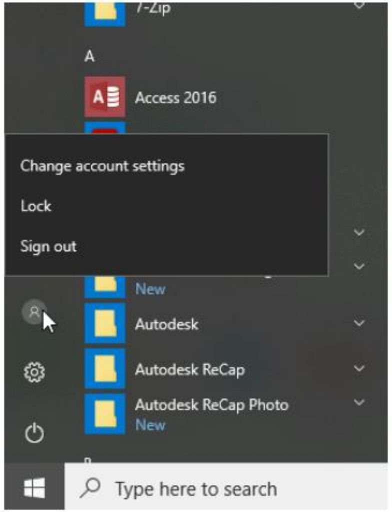
Remote Desktop Access Instructions

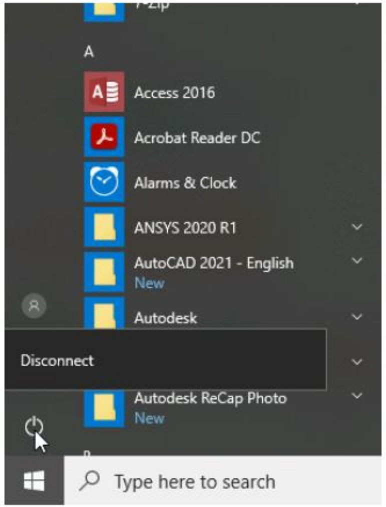


Arrangement for the tutorial on 23rd Feb

Remote Desktop Access Instructions

For general access:	If you have reserved a PC:
Go the user icon and select '**Sign out**' only to log out of remote desktop.	Go to the power icon and use '**Disconnect**' to log out of a remote session. This will ensure that your applications remain open the next time you log into the computer again.





Download Image Data and Code

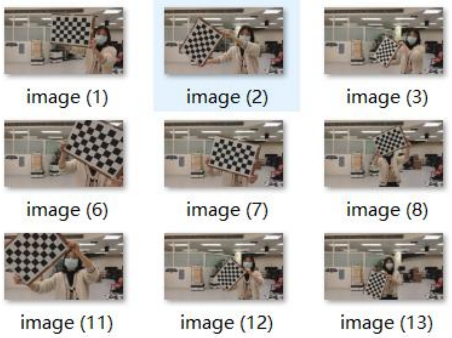
1. Connect to remote desktop via https://puuds.polyu.edu.hk/uds/page/login;
2. Download data and code at the google drive link and Extract it in the remote desktop.

https://drive.google.com/file/d/1kzwPI4-icW6AZy3HvoAMiHtdBScn7AKa/view?usp=sharing

- CalibrationImages
- StructureFromMotion
- VisualSLAM

StructureFromMotion and VisualSLAM: Two VO application examples

CalibrationImages



The images are captured by XiaoMi 8

StructureFromMotion

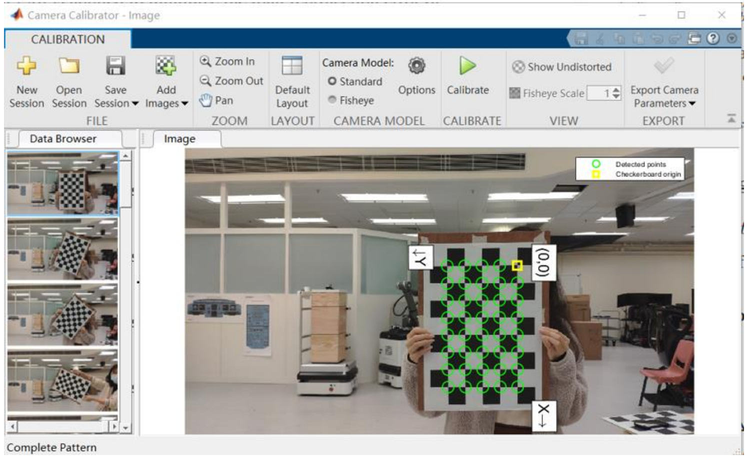
- Dataset1
- cameraParams
- getColor
- helperEstimateRelativePose
- StructureFromMotion

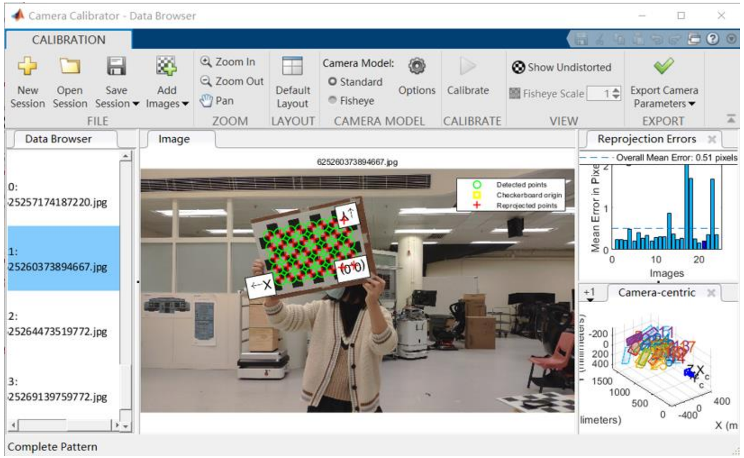
VisualSLAM

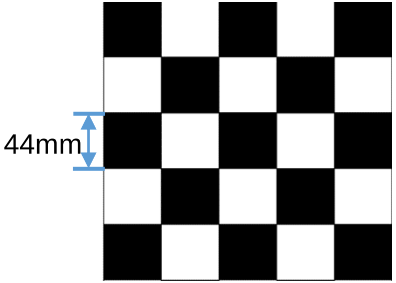
- Dataset2
- MonocularSLAM
- helperVisualizeMotionAndStructure
- helperVisualizeMatchedFeatures
- helperTriangulateTwoFrames
- helperTrackLocalMap
- helperTrackLastKeyFrame
- helperSURFFeatureExtractorFunction
- helperSelectStrongConnections
- helperMatchFeaturesInRadius

…

Camera Calibration using MATLAB







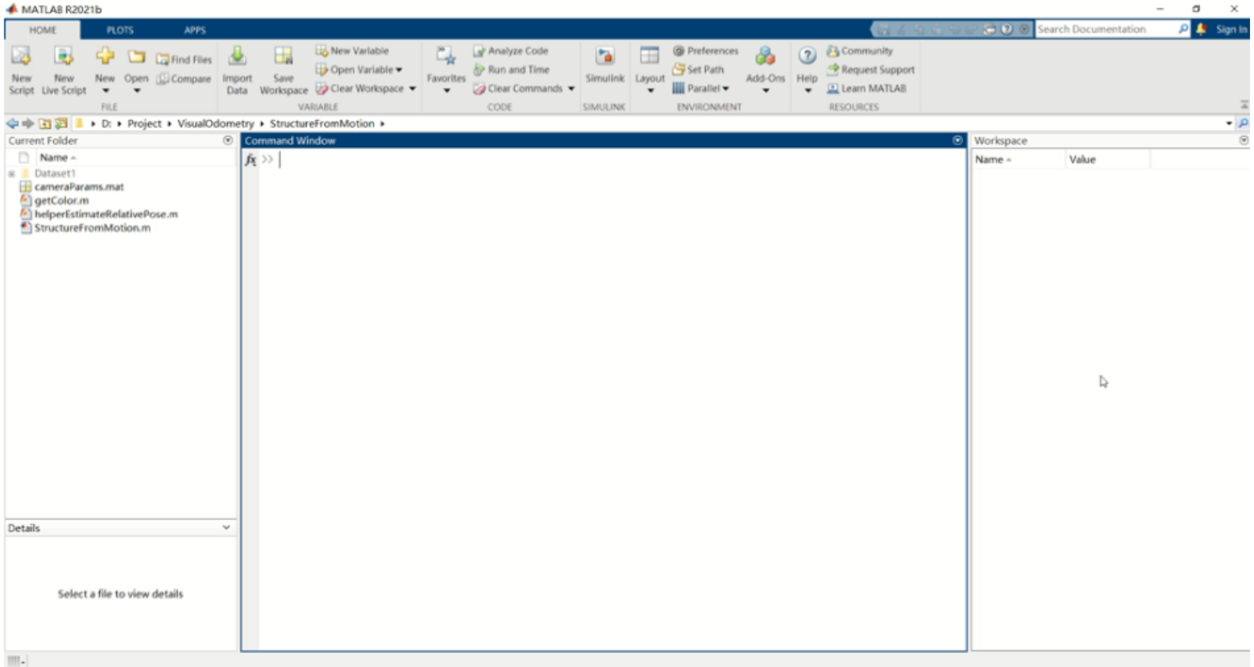
Checkerboard

cameraParams | cameraParams.Intrinsics

cameraParams.Intrinsics

属性	值
FocalLength	[1.0054e+03,1.0032e+03]
PrincipalPoint	[654.6171,370.9098]
ImageSize	[720,1280]
RadialDistortion	[0.1889,-0.3517]
TangentialDistortion	[0,0]
Skew	0
IntrinsicMatrix	[1.0054e+03,0,0;0,1.0032e+03,0;654.6171,370.9098,1]

Camera Calibration using MATLAB



The tutorial video: https://youtu.be/iT55UyLeNvs

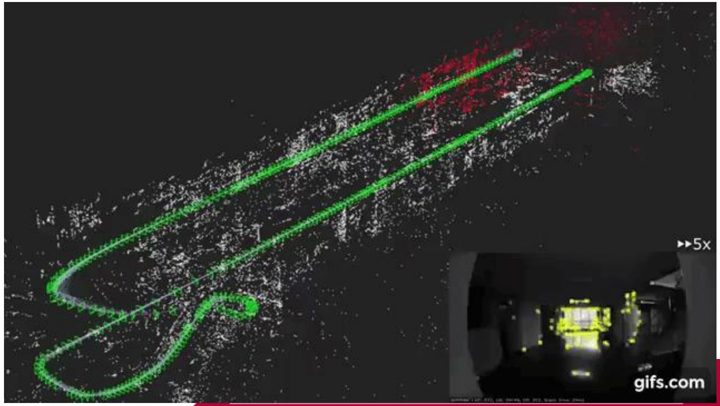
Visual odometry with a camera

Find the representative features in an image!

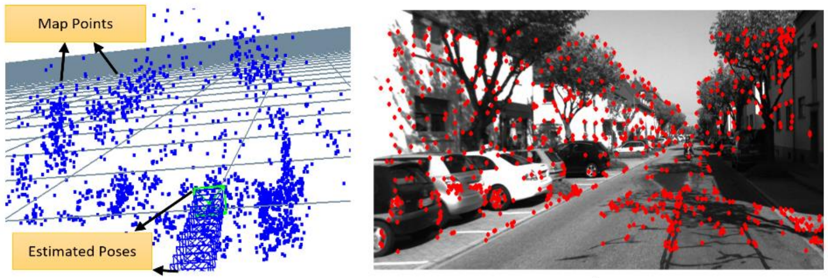
Formulate the difference between stereo and monocular visual positioning!

Camera → Image → Feature Detection → Feature Tracking → Feature Association → Pose Estimation

Find the same features in consecutive image!



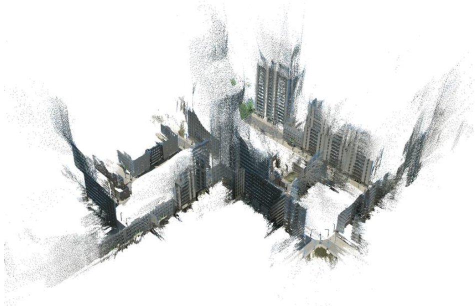
Visual Odometry (VO) Using MATLAB



(a) (b)

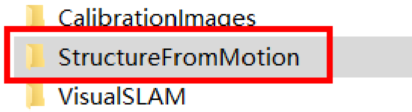
The two VO examples

- Structure From Motion (SFM): This example shows you how to estimate the poses of a calibrated camera from a sequence of views, and reconstruct the 3-D structure of the scene. (Dense features, time-consuming!) Popular in photogrammetry!
- Visual SLAM/odometry: From a monocular camera to build a map of an indoor environment and estimate the trajectory of the camera. (Sparse features, efficient!) Popular in Robotics navigation!

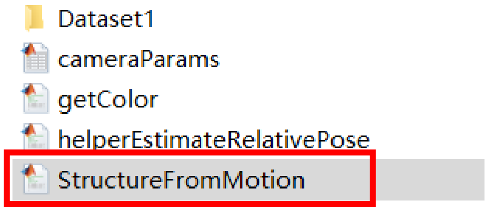


Experiment 1: Structure From Motion

The code and dataset are in the StructureFromMotion folder

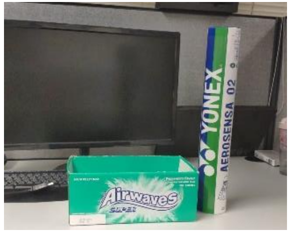
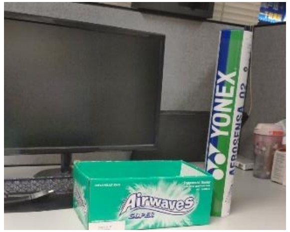


The main function is the StructureFromMotion.m
Open it using MATLAB



There is a set of images captured from different views stored in Dataset1



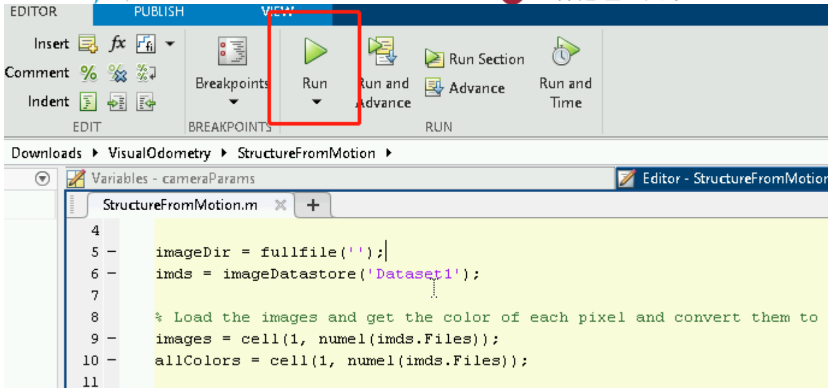




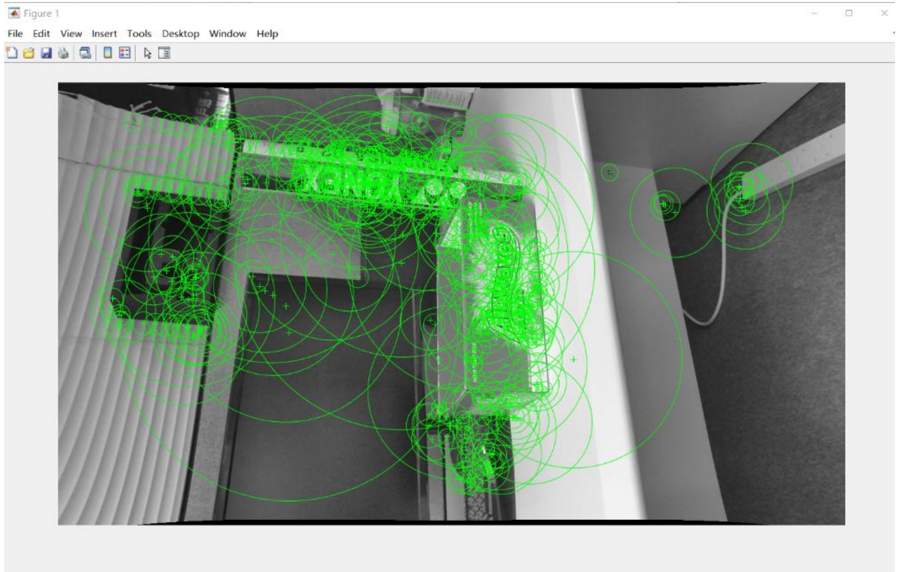
Run it

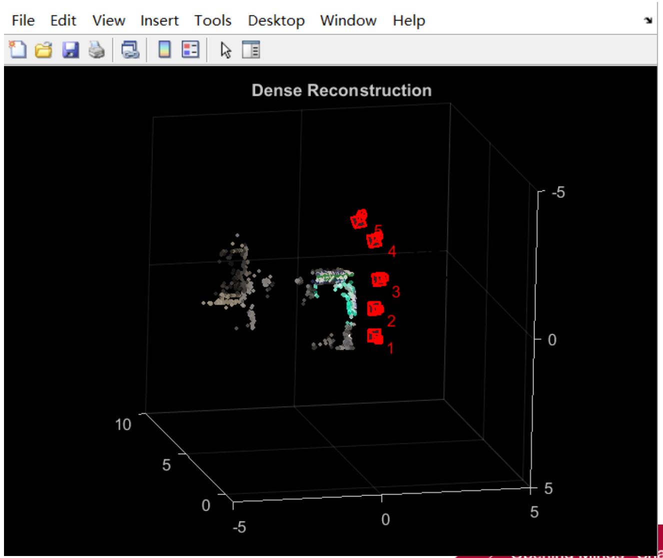
Experiment 1: Structure From Motion

Change the file path to yours, make sure the function can get the Datasets and cameraParam.mat

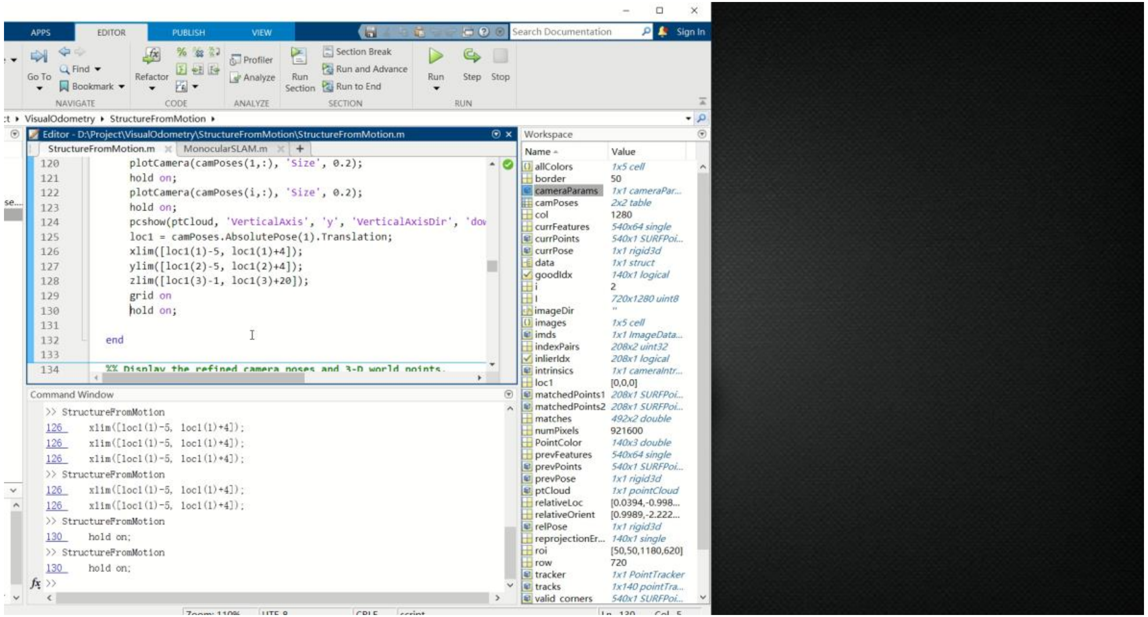
```
imageDir = fullfile('');
imds = imageDatastore('Dataset1');
```

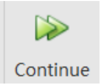






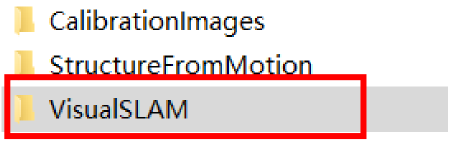
Experiment 1: Structure From Motion



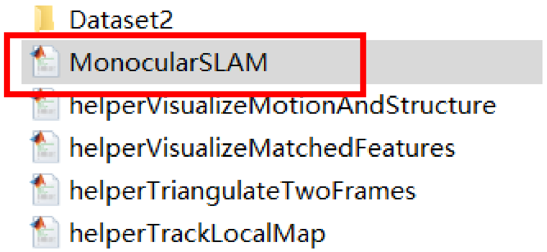
If you want to observe the results step by step, you can set a breakpoint at line 130 of the main function, and then press 

Experiment 2: Monocular Visual SLAM

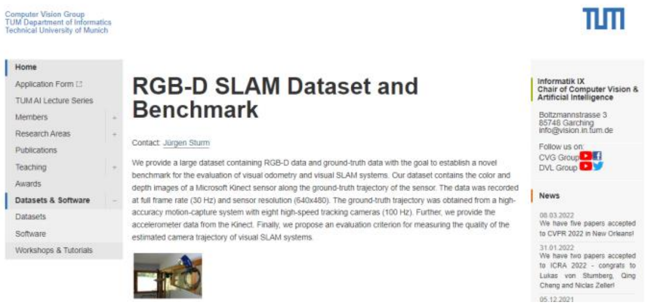
The code and dataset are in the VisualSLAM folder

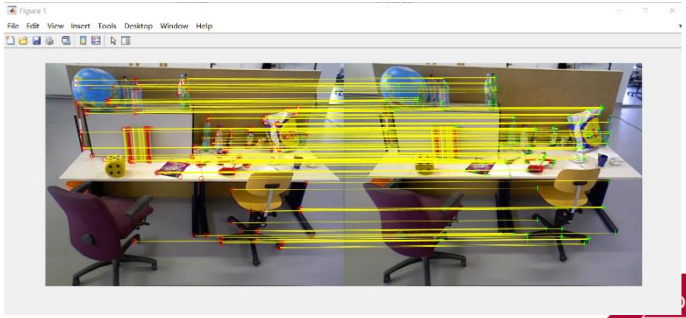


The main function is the MonocularSLAM.m
Open it using MATLAB



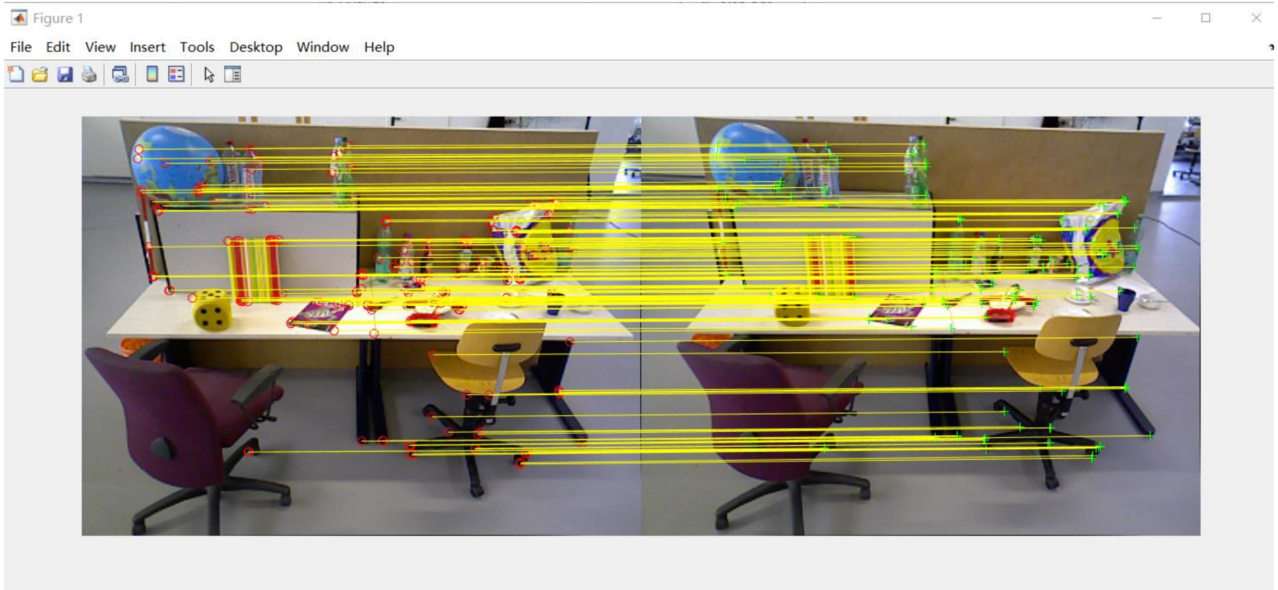
The data (Dataset1) used in this example are from the [TUM RGB-D benchmark](), which is a public dataset, and we select a segment of it for demonstration.





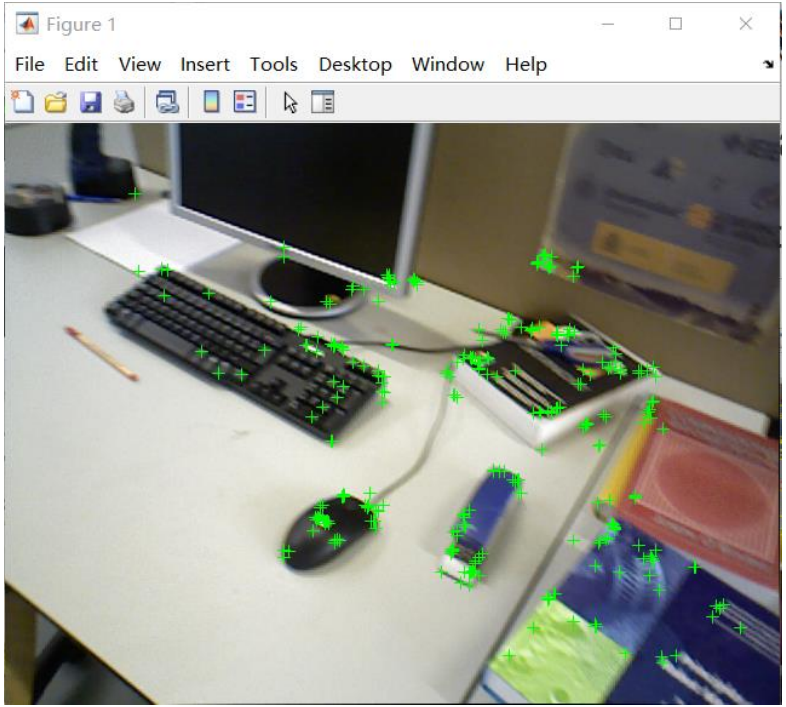
Experiment 2: Monocular Visual SLAM

The output result:

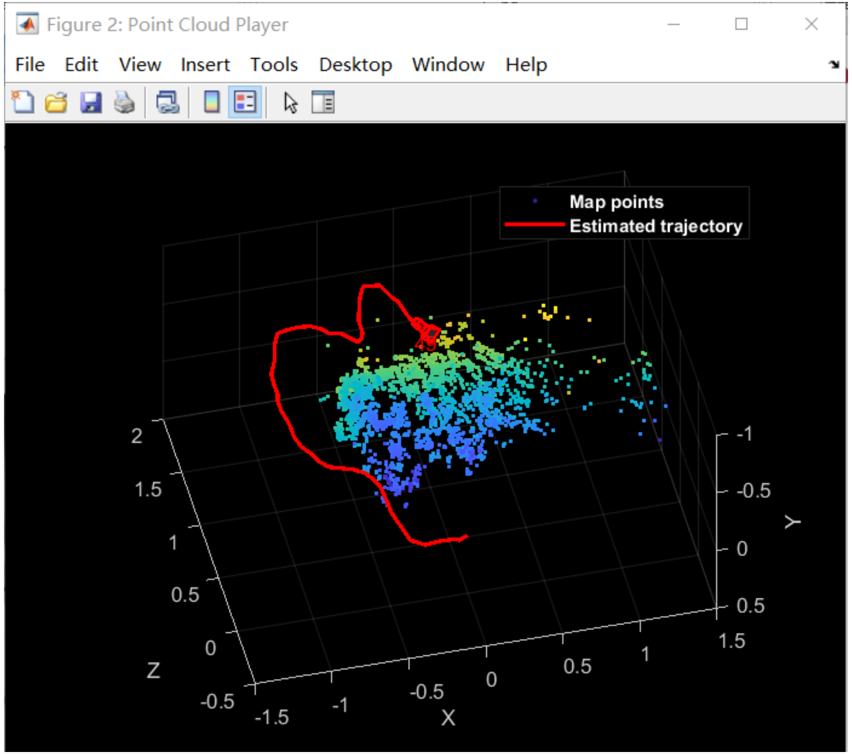


The ORB feature detection and matching result.

Experiment 2: Monocular Visual SLAM

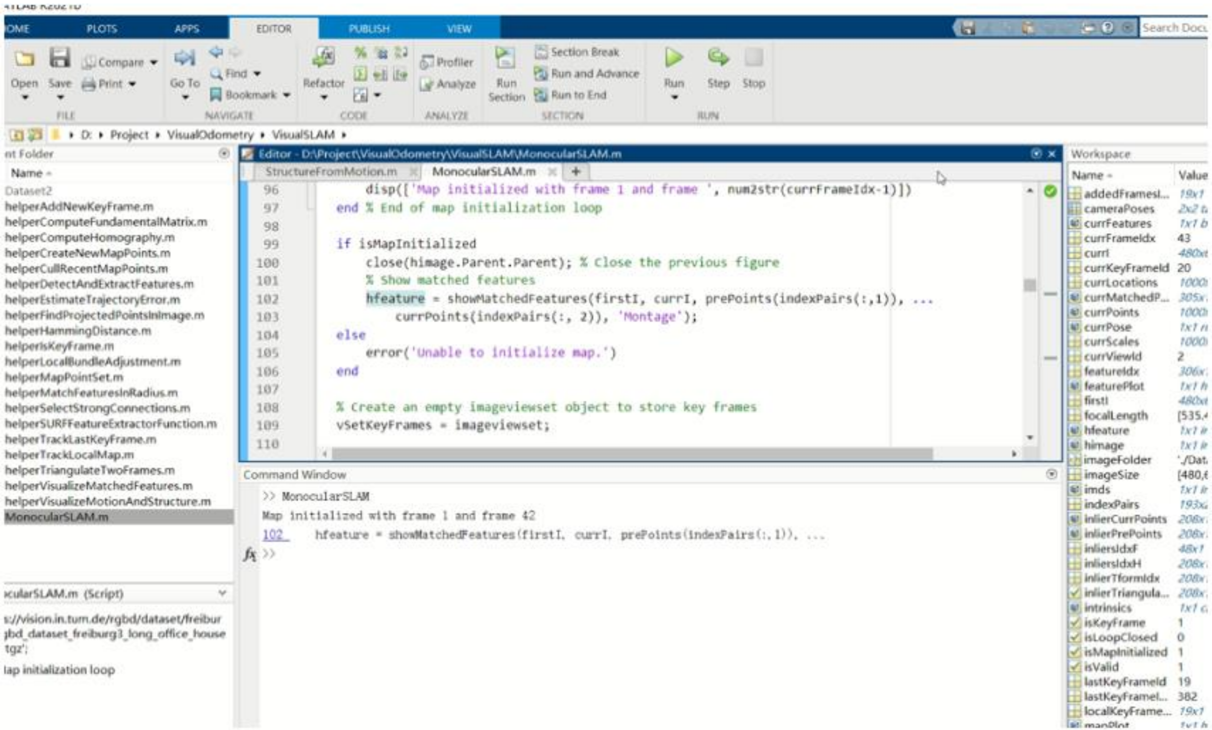


The feature detection



The estimated trajectory

Experiment 2: Monocular Visual Odometry/SLAM



The demo (no voice): https://youtu.be/ksMr1fiKwbY

Q&A

Thank you for your attention ☺

Q&A

Thank you very much the help from teaching assistant (Xi Zheng and Pin Hsun Lee)