

# 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*

# 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

<p><b>Zhang, Ji, and Sanjiv Singh. "Low-drift and real-time lidar odometry and mapping." <i>Autonomous Robots</i> 41, no. 2 (2017): 401-416.</b></p>	<p><b>Qin, Tong, Peiliang Li, and Shaojie Shen. "Vins-mono: A robust and versatile monocular visual-inertial state estimator." <i>IEEE Transactions on Robotics</i> 34.4 (2018): 1004-1020.</b></p>
<p>Wen, Weisong, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. "Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter." <i>NAVIGATION, Journal of the Institute of Navigation</i> 68, no. 2 (2021): 315-331.</p>	<p>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." <i>IEEE Transactions on Robotics</i> 37, no. 6 (2021): 1874-1890.</p>
<p>Wen, W. and Hsu, L.T., 2021, May. Towards robust GNSS positioning and Real-time kinematic using factor graph optimization. In <i>2021 IEEE International Conference on Robotics and Automation (ICRA)</i> (pp. 5884-5890). IEEE.</p>	<p>Engel, Jakob, Vladlen Koltun, and Daniel Cremers. "Direct sparse odometry." <i>IEEE transactions on pattern analysis and machine intelligence</i> 40, no. 3 (2017): 611-625.</p>
<p>Navigation Technique of Tesla's Autonomous Driving</p>	<p>Navigation Technique of Google's (now Waymo) Autonomous Driving</p>
<p>GNSS Real-time Kinematic Positioning for Autonomous Driving</p>	<p>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 <i>2020 IEEE International Conference on Robotics and Automation (ICRA)</i> (pp. 4322-4328). IEEE.</p>

# 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

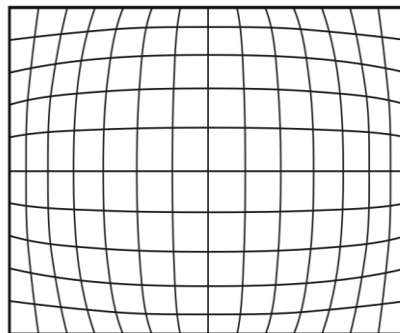
## Distortion Reasons

### Why distortion occur?

- Optical distortion
- Assembly of the camera

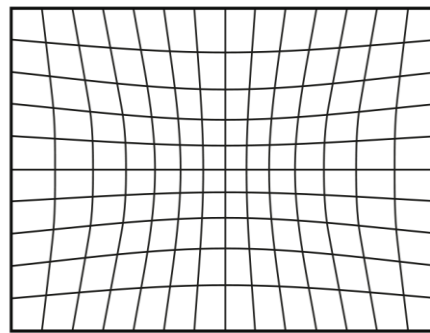
Due to **lens shape**,  
called **radial distortion**

## Different degree of distortion



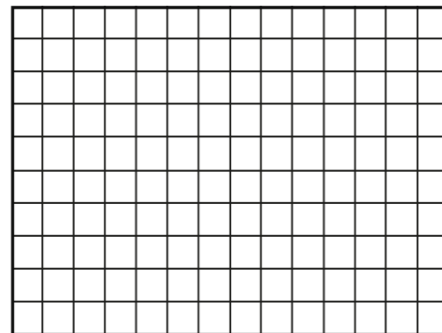
Barrel distortion

Severe distortion in the middle



Pincushion distortion

Severe distortion in the boundary



No distortion



# Model of the camera

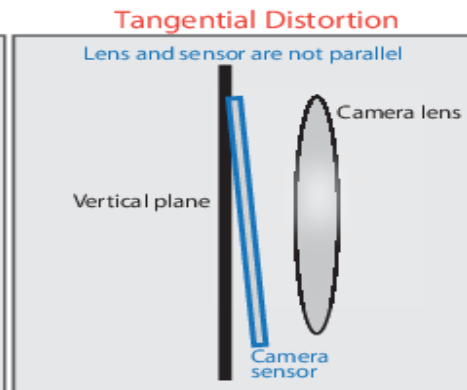
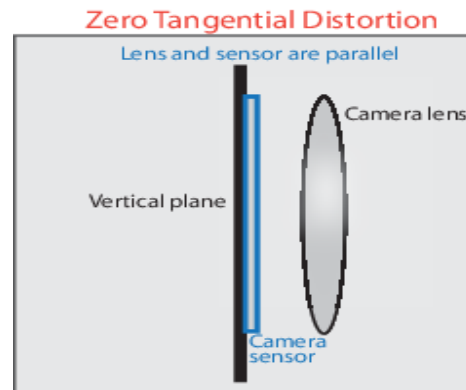
## Distortion Reasons

### 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

$$x_c = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2)$$

$$y_c = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy$$

$k_1$ ,  $k_2$ , and  $k_3$  — Radial distortion coefficients of the lens

$p_1$  and  $p_2$  — Tangential distortion coefficients of the lens

$r^2$ :  $x^2 + y^2$

## Illustration



Original Photo



Corrected photo

# Model of the camera

## Calibration **correction**

$$x_c = x(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1xy + p_2(r^2 + 2x^2)$$

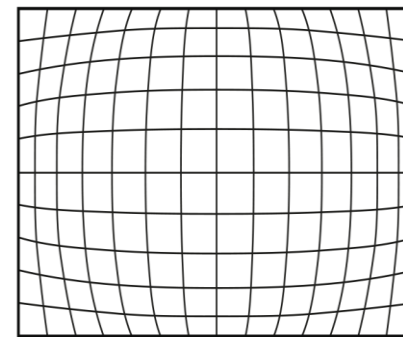
$$y_c = y(1 + k_1r^2 + k_2r^4 + k_3r^6) + p_1(r^2 + 2y^2) + 2p_2xy$$

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

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

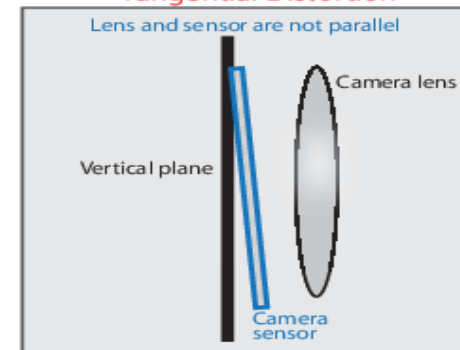
**p1 and p2** — Tangential distortion coefficients of the lens

How to calibrate ? And how  
many parameters to calibrate?



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

**Tangential Distortion**



**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**

Items	param1	param2	param3	param4	param5
Camera Intrinsic-K	$f_u$	$f_v$	$\Delta u$	$\Delta v$	
Lens Distortion	K1	K2	K3	p1	p2

# Camera Calibration

## Calibration of camera

Algorithm: Zhang Zhengyou Calibration<sup>[1]</sup>



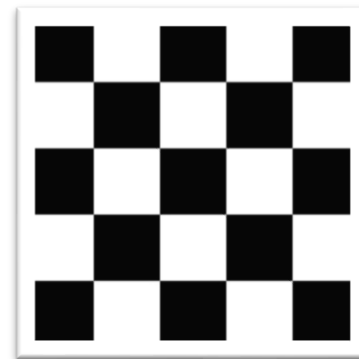
He received the IEEE Helmholtz Time Test Award for "Zhang's Calibration Method" in 2013

A very famous expert in computer vision and multimedia technology

### Advantages:

The equipment is simple, just a printed checkerboard;

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

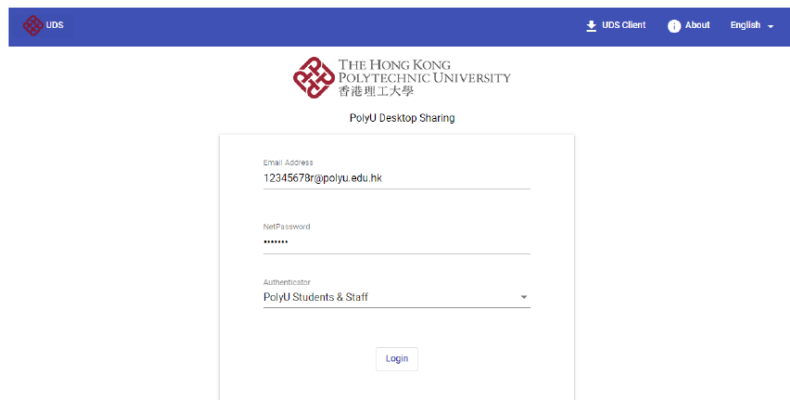


[1] Zhang, Zhengyou. "A flexible new technique for camera calibration." *IEEE Transactions on pattern analysis and machine intelligence* 22.11 (2000): 1330-1334.

# 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>)

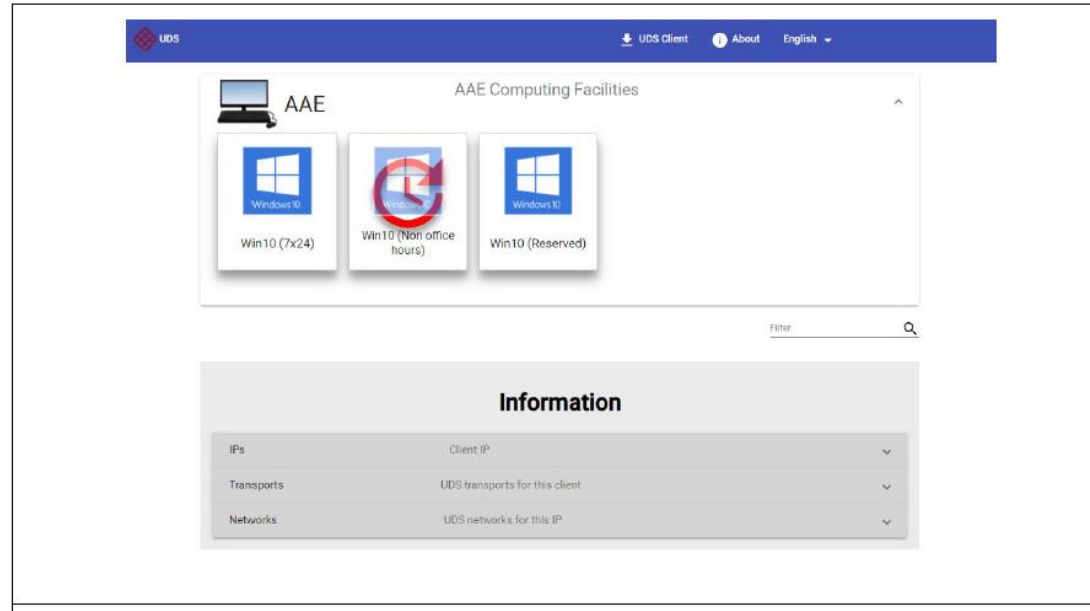


2. For general access outside of office hours, click on the "Win10 (Non office hours)" option. It will only be available when the lab is physically closed.  
If you have reserved a PC, the "Win10 (Reserved)" option will be available to you.

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

# Arrangement for the tutorial on 23rd Feb

## Remote Desktop Access Instructions

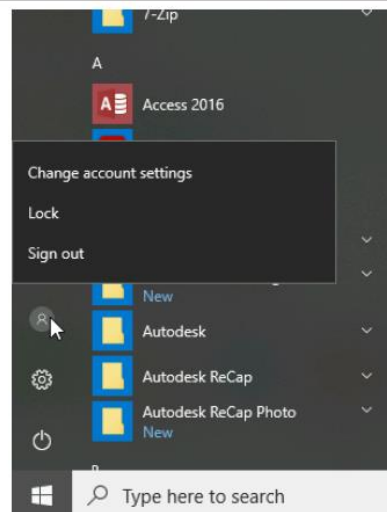


# Arrangement for the tutorial on 23rd Feb

## Remote Desktop Access Instructions

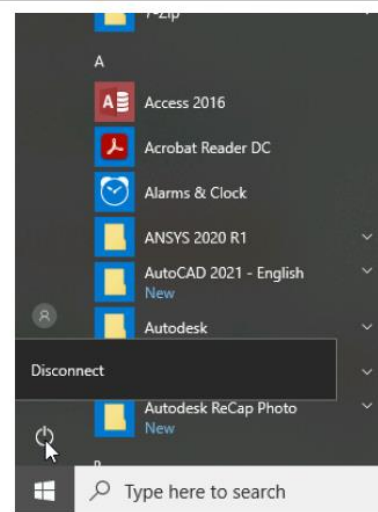
### For general access:

Go to the user icon and select **'Sign out' only** to log out of remote desktop.



### If you have reserved a PC:

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>

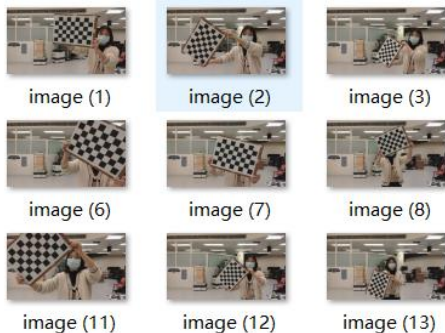
CalibrationImages

StructureFromMotion

VisualSLAM

} 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

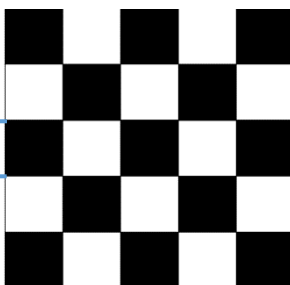
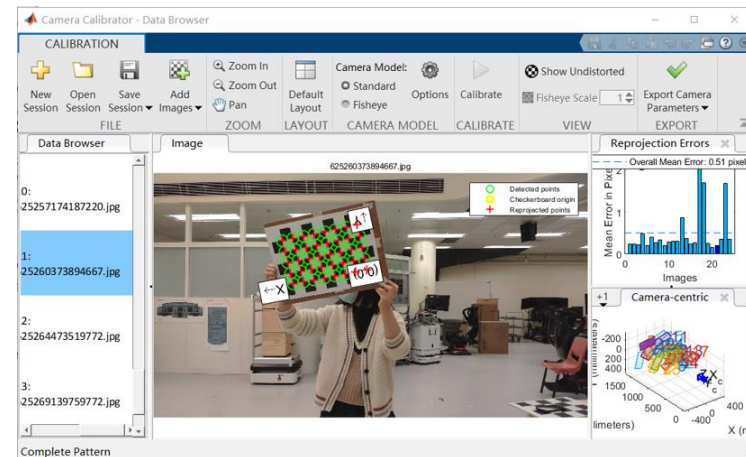
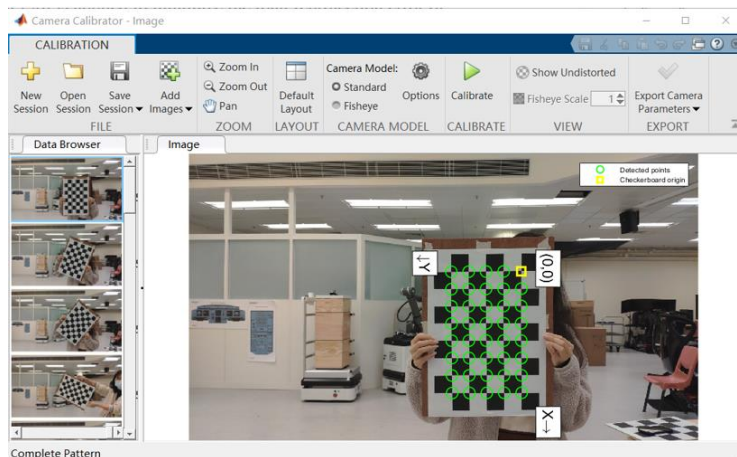
helperSURFFeatureExtractorFunction

helperSelectStrongConnections

helperMatchFeaturesInRadius

...

# Camera Calibration using MATLAB

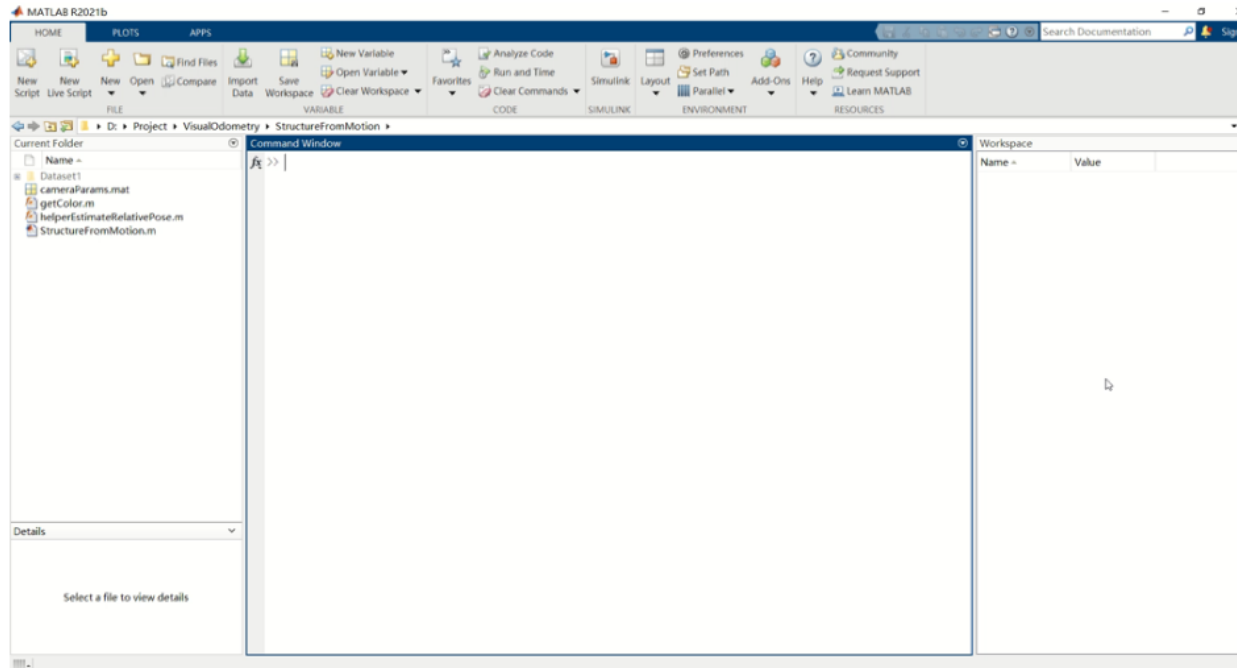


Checkerboard



cameraParams	
cameraParams.Intrinsics	
cameraParams.Intrinsics	
屬性	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



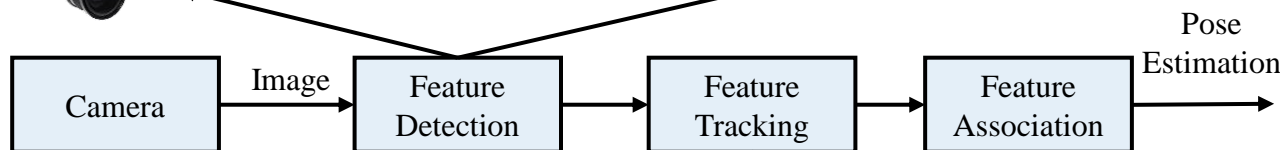
The tutorial video: <https://youtu.be/iT55UyLeNvs>

# Visual odometry with a camera

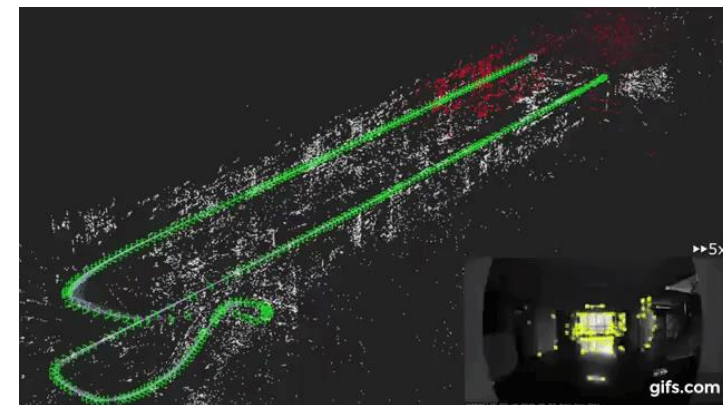
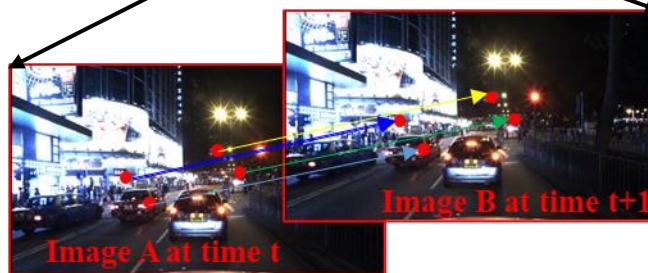


Find the representative features in an image!

Formulate the difference between stereo and monocular visual positioning!



Find the same features in consecutive image!

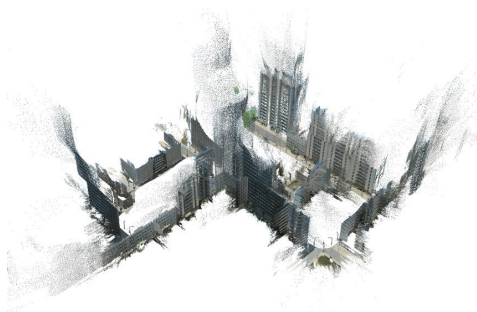
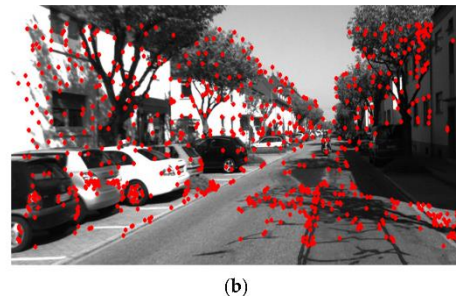
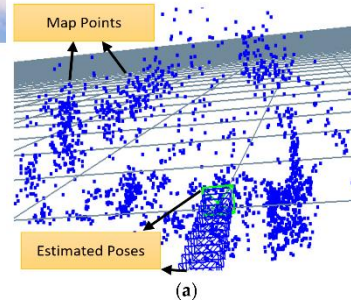




# Visual Odometry (VO) Using MATLAB

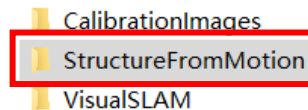
## The two VO examples

- 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!

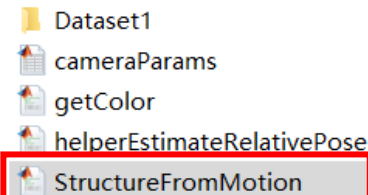


## Experiment 1: Structure From Motion

The code and dataset are in the StructureFromMotion folder



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



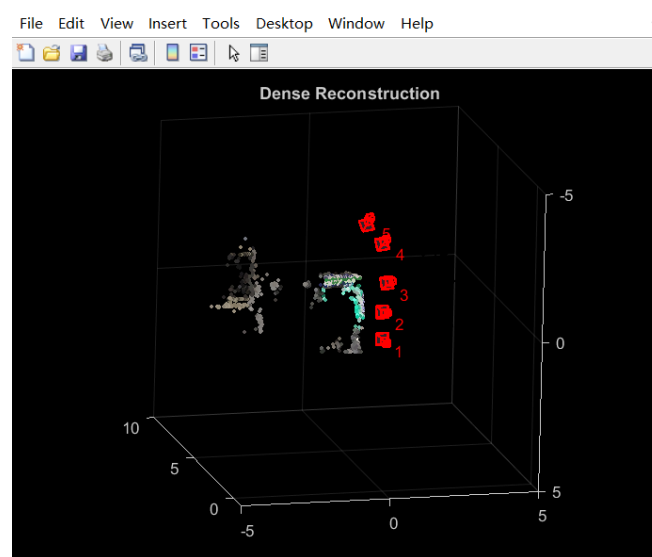
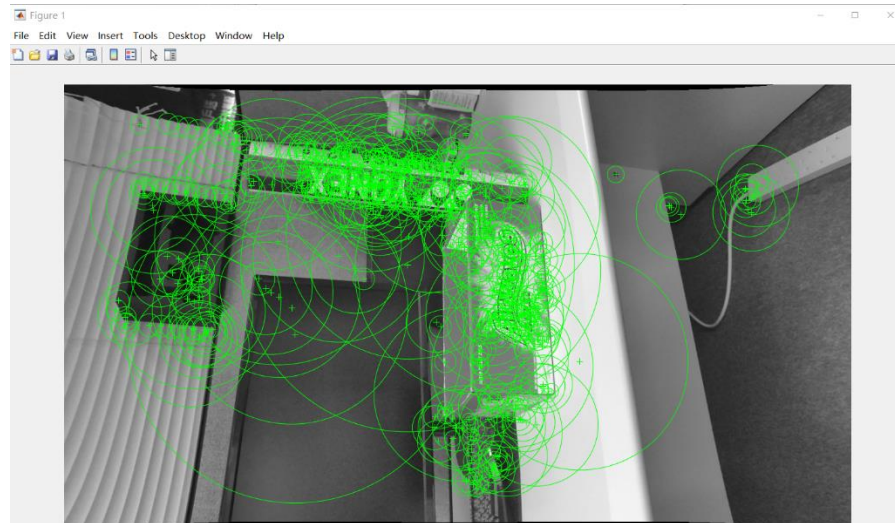
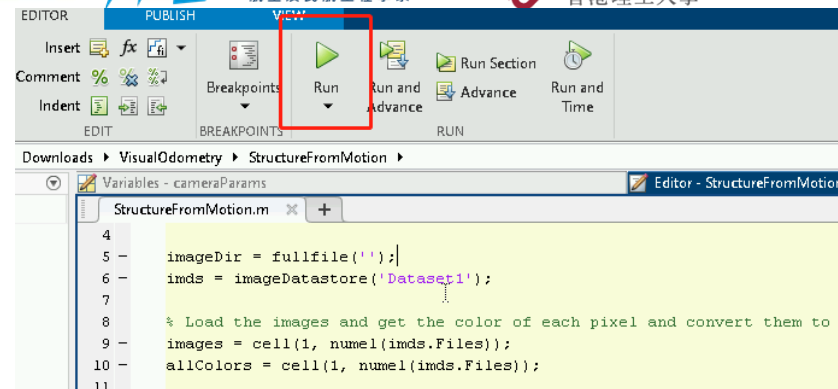
There is a set of images captured from different views stored in  
Dataset1



# Experiment 1: Structure From Motion

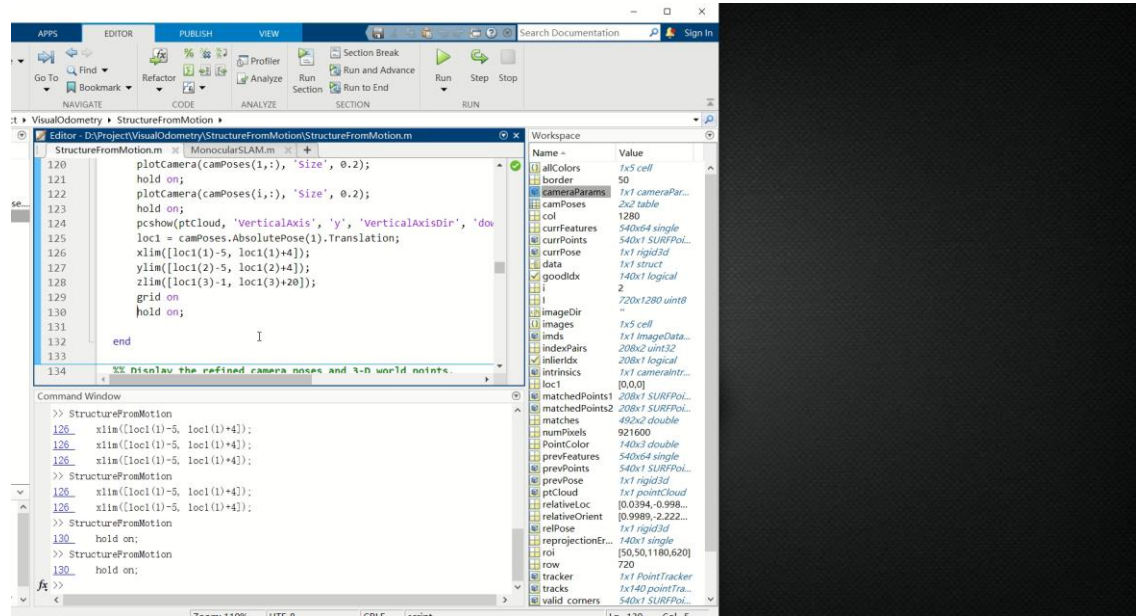
Change the file path to yours, make sure the function can get the Datasets and cameraParam.mat

```
4
5 imageDir = fullfile('');
6 imds = imageDatastore('Dataset1');
7
```





# 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

CalibrationImages  
StructureFromMotion  
**VisualSLAM**

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

Dataset2  
**MonocularSLAM**  
helperVisualizeMotionAndStructure  
helperVisualizeMatchedFeatures  
helperTriangulateTwoFrames  
helperTrackLocalMap

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.

Computer Vision Group  
TUM Department of Informatics  
Technical University of Munich

TUM

### RGB-D SLAM Dataset and Benchmark

Contact: [Jürgen Sturm](#)

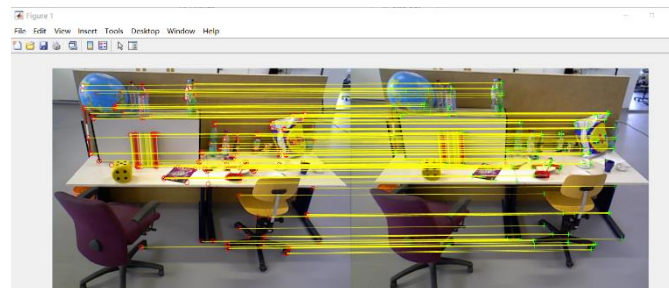
We provide a large dataset containing RGB-D data and ground truth data with the goal to establish a novel benchmark for the evaluation of visual odometry and visual SLAM systems. Our dataset contains the color and depth images of a Microsoft Kinect sensor along the ground-truth trajectory of the sensor. The data was recorded at full frame rate (30 Hz) and sensor resolution (540x480). The ground-truth trajectory was obtained from a high-accuracy motion-capture system with eight high-speed tracking cameras (100 Hz). Further, we provide the accelerometer data from the Kinect. Finally, we propose an evaluation criterion for measuring the quality of the estimated camera trajectory of visual SLAM systems.



Informatic IX  
Chair of Computer Vision &  
Artificial Intelligence  
Boltzmannstrasse 3  
85748 Garching  
info@vision.in.tum.de

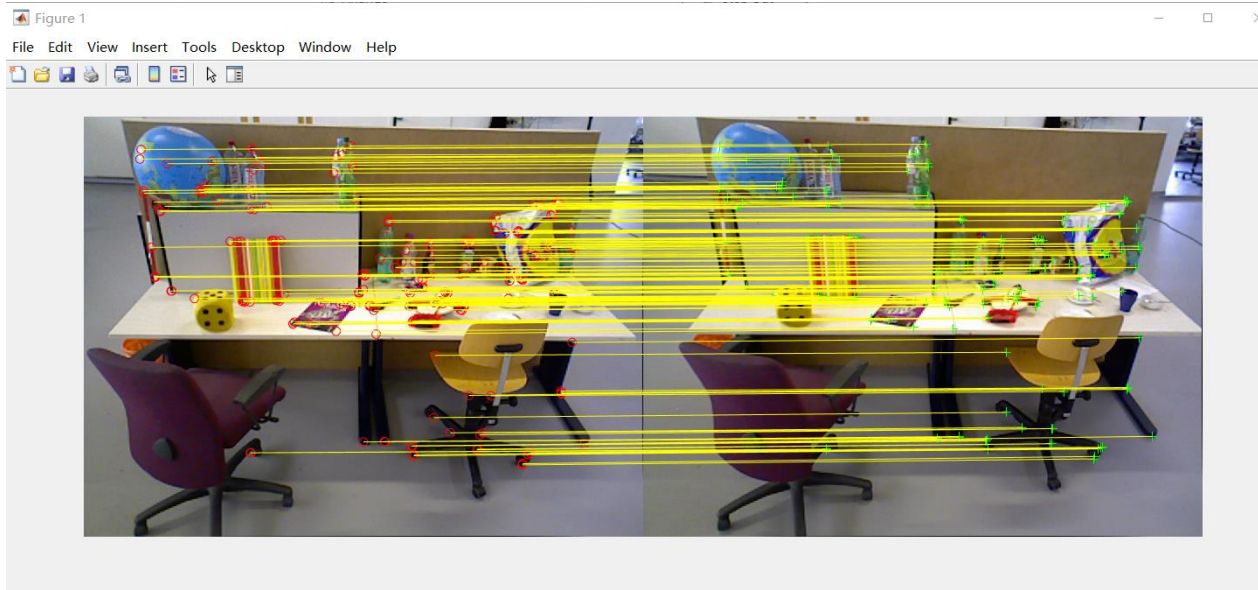
Follow us on:  
CVG Group  
DVL Group

News  
08.03.2022  
We have five papers accepted to CVPR 2022 in New Orleans!  
31.01.2022  
We have two papers accepted to ICRA 2022 - congrats to Lukas von Shubert, Qing Cheng and Niklas Zeller!  
05.12.2021



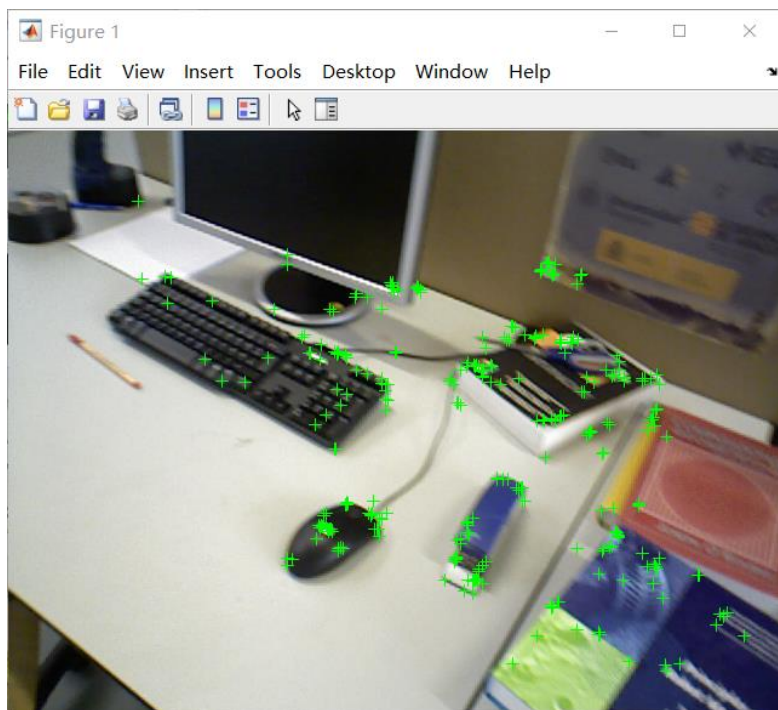
## Experiment 2: Monocular Visual SLAM

The output result:

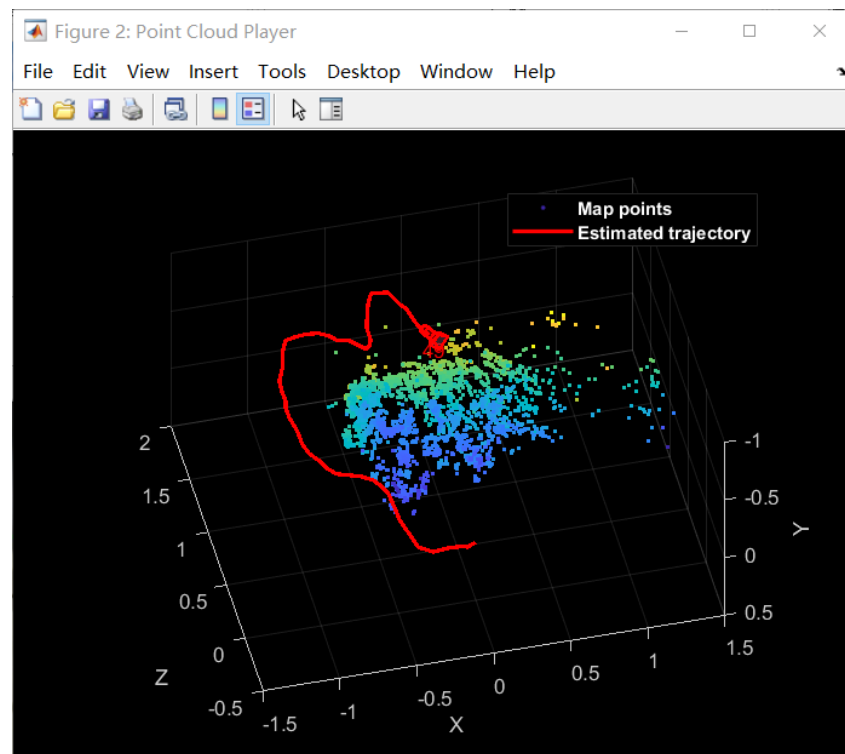


The ORB feature detection and matching result.

## Experiment 2: Monocular Visual SLAM



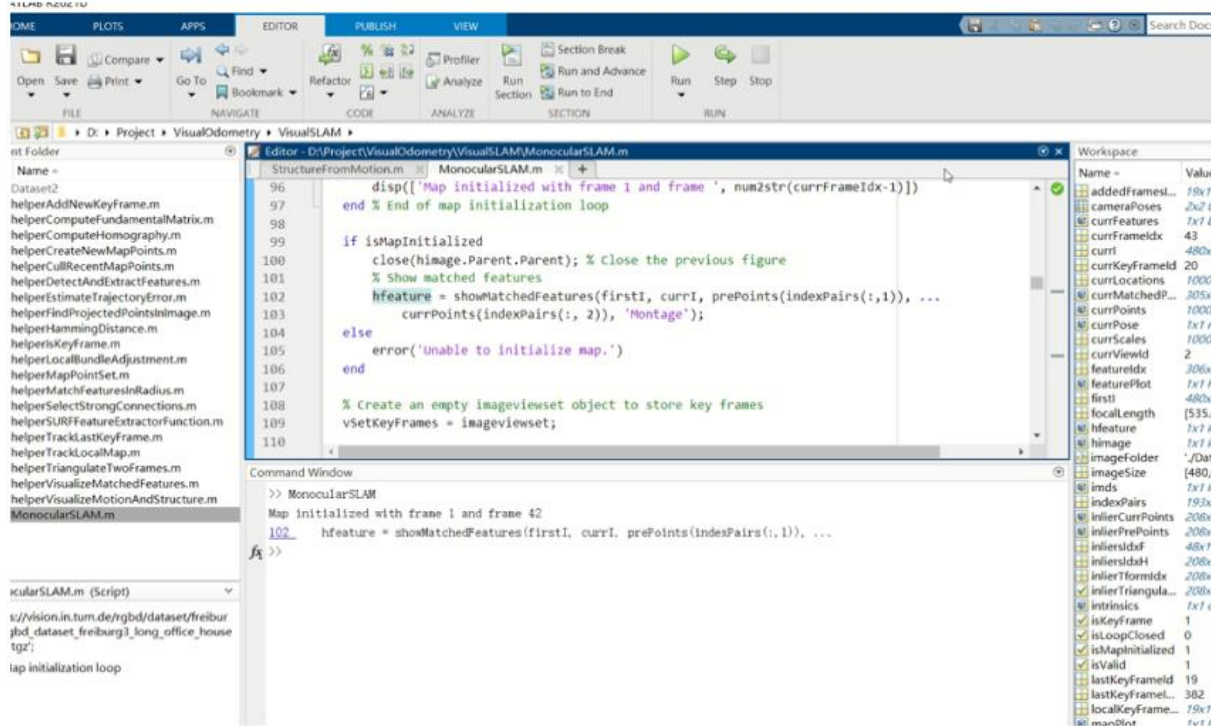
The feature detection



The estimated trajectory



## Experiment 2: Monocular Visual Odometry/SLAM



The screenshot shows the MATLAB R2019b environment. The Editor window displays the `MonocularSLAM.m` script. The Command Window shows the execution progress, including the message "Map initialized with frame 1 and frame 42". The Workspace window lists various variables and their sizes.

```

96     disp(['Map initialized with frame 1 and frame ', num2str(currFrameIdx-1)])
97   end % End of map initialization loop
98
99   if isMapInitialized
100     close(himage.Parent.Parent); % Close the previous figure
101     % Show matched features
102     hfeature = showMatchedFeatures(firstI, currI, prePoints(indexPairs(:,1)), ...
103     currPoints(indexPairs(:,2)), 'Montage');
104   else
105     error('unable to initialize map.')
106   end
107
108   % Create an empty imageviewset object to store key frames
109   vSetKeyFrames = imageviewset;
110
Command Window
>> MonocularSLAM
Map initialized with frame 1 and frame 42
102     hfeature = showMatchedFeatures(firstI, currI, prePoints(indexPairs(:,1)), ...
fx >>

Workspace
Name      Value
addedFrames 19x1
cameraPoses 2x2 6
currFeatures 1x1 6
currFrameIdx 43
currI       480x1
currKeyFrameId 20
currLocations 1000
currMatchedP... 305x
currPoints 1000
currPose    1x1 n
currScales 1000
currViewId  2
featureIdx  306x
featurePlot 1x1 6
firstI     480x1
focalLength 535.4
hfeature    1x1 6
himage     1x1 6
imageFolder './Dat
imageSize   480,f
imds        1x1 6
indexPairs 193x6
inlierCurrPoints 208x
inlierPrePoints 208x
inliersdbH  48x1
inliersdbH  208x
inlierFormIdx 208x
inlierTriangula... 208x
intrinsic   1x1 6
isKeyFrame  1
isLoopClosed 0
isMapInitialized 1
isValid     1
lastKeyFrameId 19
lastKeyFrame... 382
localKeyFrame... 19x1
mandPlot    1x1 6
  
```

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)