Stereo Vision and Visual Odometry

A naïve implementation from scratch in C++

M.Sc. Wang, Yu

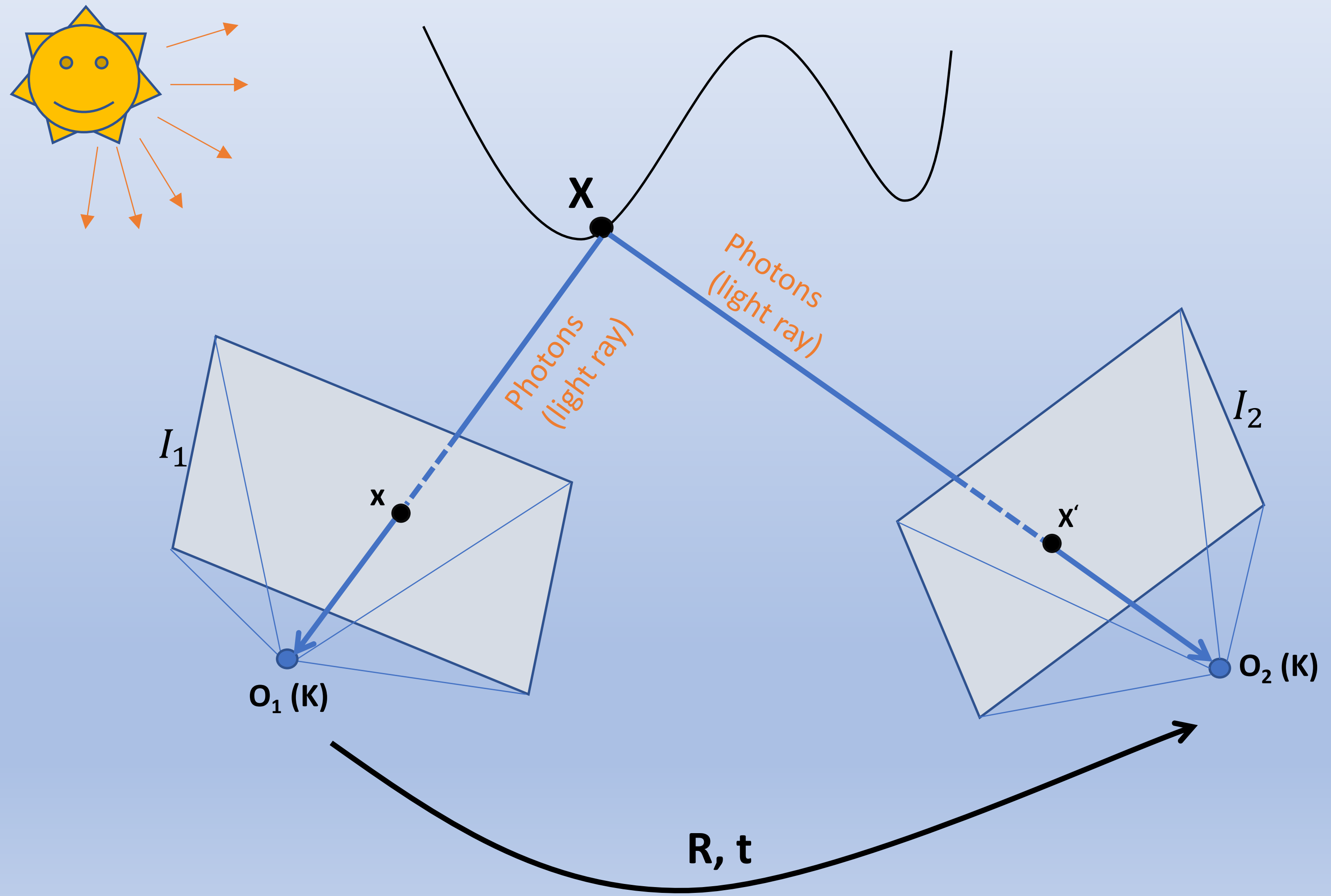
Overview

- Introduction
- Stereo - from 2D to 3D
- Tracking via Optimisation
- Results
- Appendix

Overview

- Introduction
- Stereo - from 2D to 3D
- Tracking via Optimisation
- Results
- Appendix

Introduction - Problem



Inputs(known): Images

Estimate(unknown):
- 3D structure X
- Camera Motion [R|t]

Math Model (photo-consistency):

$$I_1(x) \triangleq I_2(x')$$

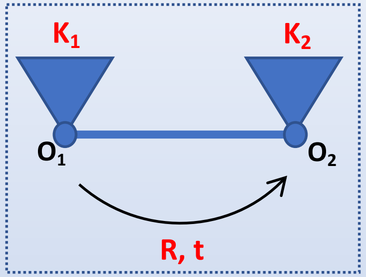
$$I_1(x) \triangleq I_2(\pi(R\pi^{-1}(x, d_x, K) + t))$$

Solution (stereo & optimisation)
- d_x solved via stereo
- [R|t] solved via non-linear Opt.:

$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_x [I_2(\phi(x)) - I_1(x)]^2$$

Introduction - Pipeline

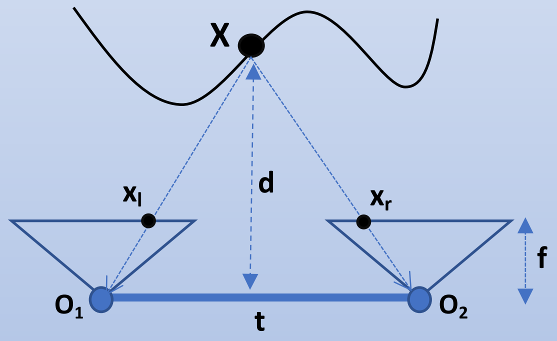
Calibration



Kalibr [1]

OpenCV [4]

Disparith/Depth

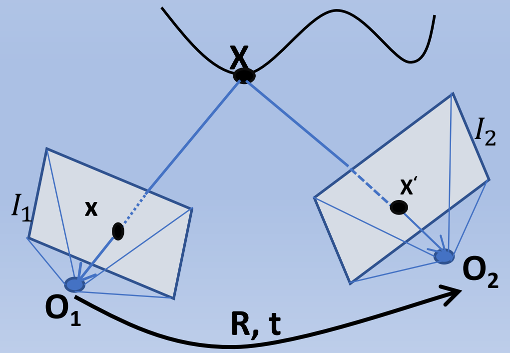


$$d = \frac{f \cdot t}{x_l - x_r}$$

Middlebury Stereo Datasets [2]

intel Intrinsics SSE [3]

Tracking/Motion



$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_x [I_2(\phi(x)) - I_1(x)]^2$$

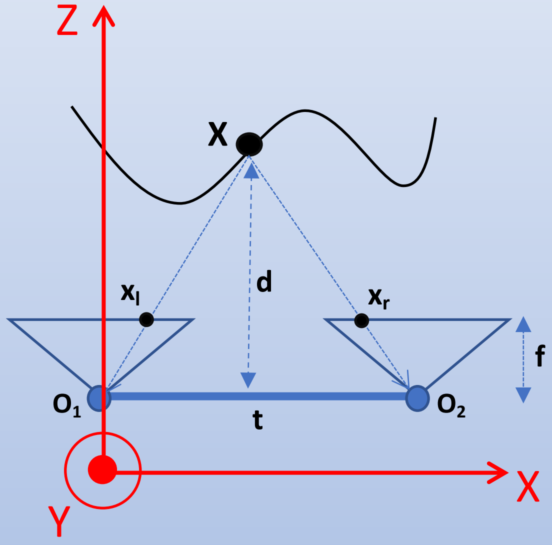
Eigen [5]

Sophus - Lie groups [6]

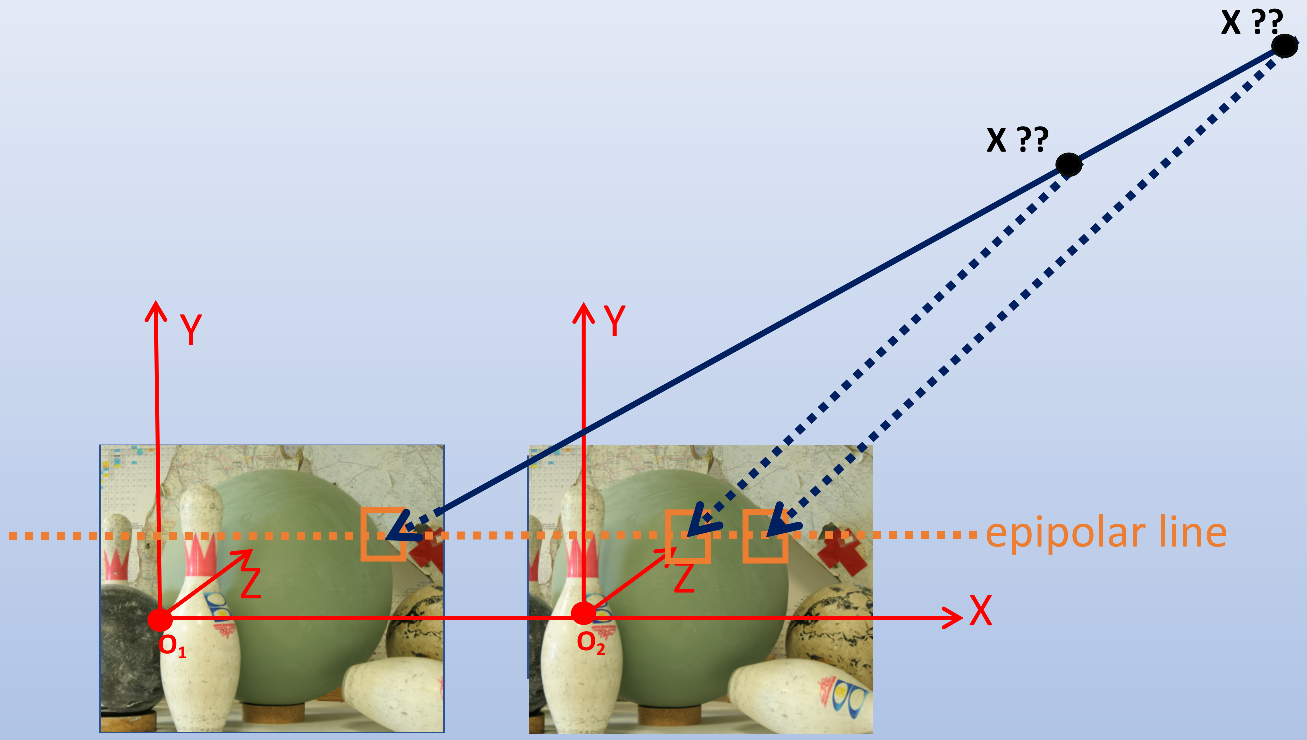
Overview

- Introduction
- Stereo - from 2D to 3D
- Tracking via Optimisation
- Results
- Appendix

Stereo - from 2D to 3D



$$d = \frac{f \cdot t}{x_l - x_r}$$



Left Image[2]

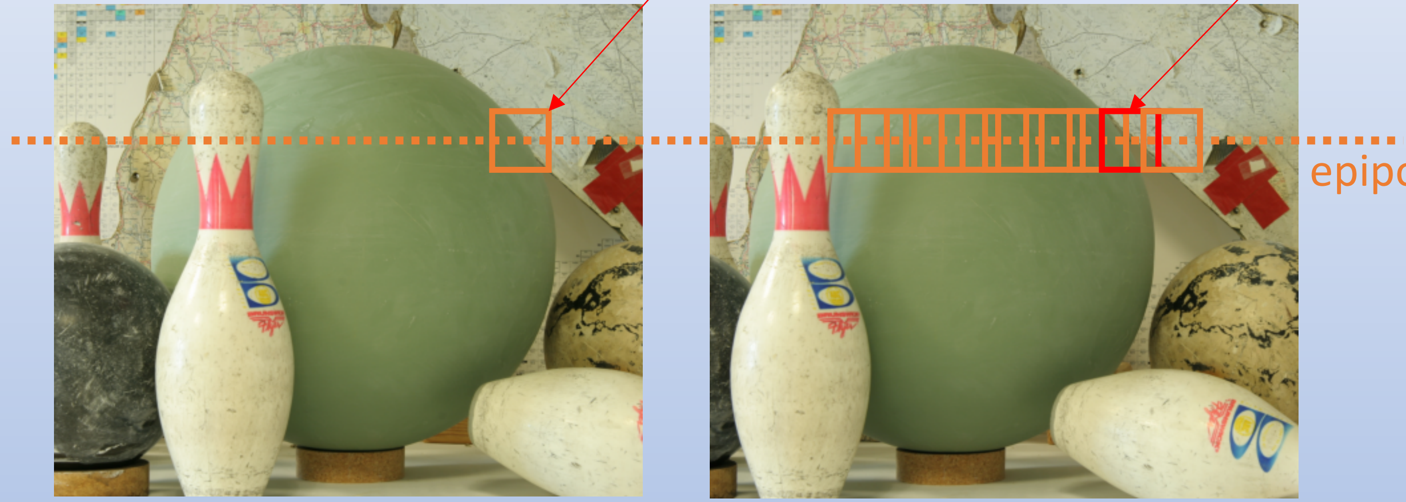
Right Image[2]

Disparity Search
(along epipolar line)

Stereo - from 2D to 3D

Template centered at x

Best match centered at x'

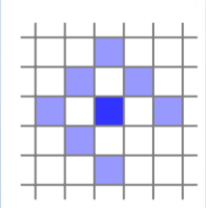


epipolar line

Left Image[2]

Right Image[2]

Implementation Consideration



Window pattern[7]

- Robust pattern
- 8 x 32 = 256 bit

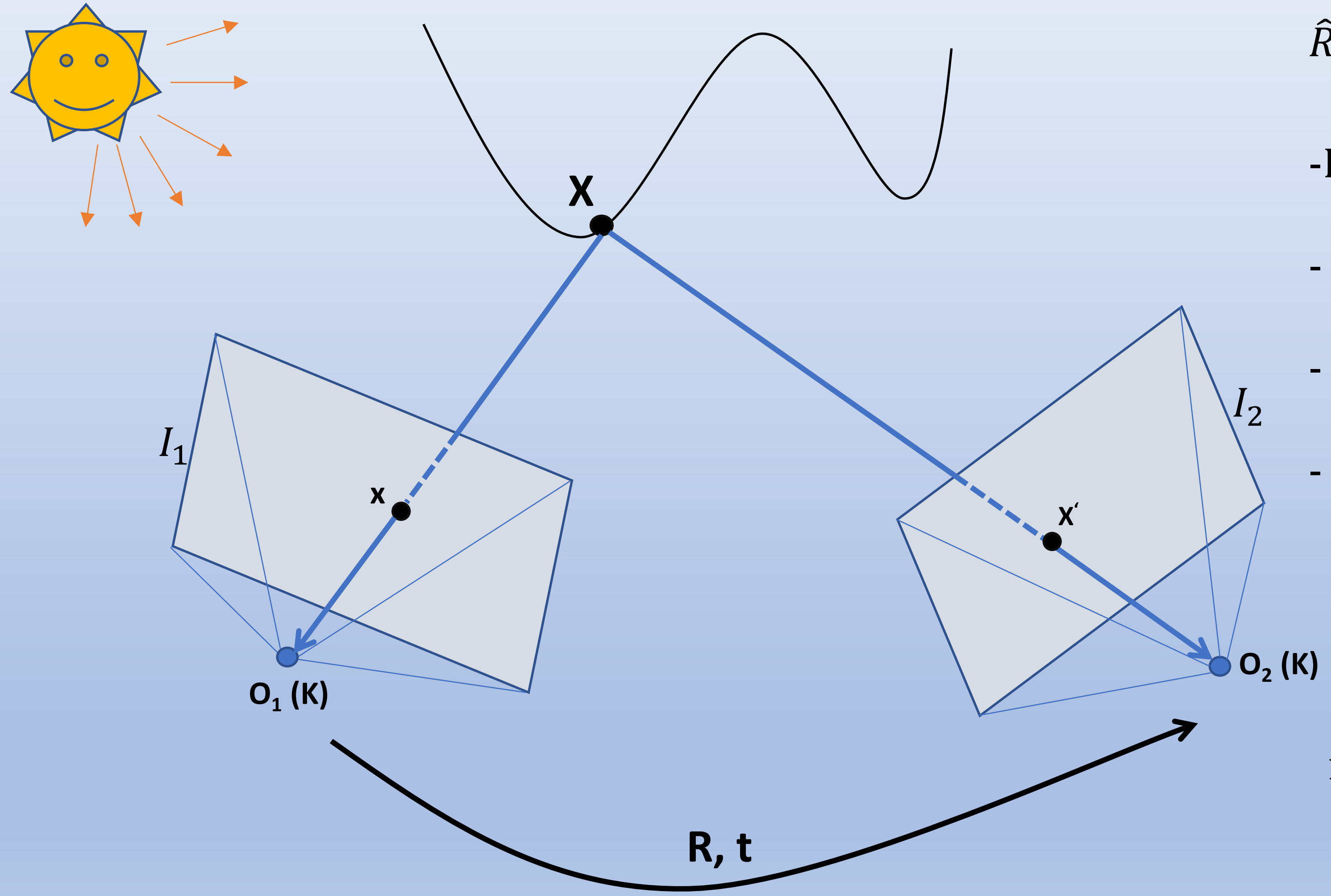
fit to SSE register

$$SSD(x, x') = \sum_{u \in window} [I_l(x-u) - I_r(x'-u)]^2$$

Overview

- Introduction
- Stereo - from 2D to 3D
- Tracking via Optimisation
- Results
- Appendix

Tracking via Optimisation



$$\hat{R}, \hat{t} = \arg\min_{R,t} \sum_x [I_2(\phi(x)) - I_1(x)]^2$$

-Pixel to 3d Camera Coord in O_1

$$\pi^{-1}(x, d_x, K)$$

- From Camera Coord in O^1 to O^2

$$R\pi^{-1}(x, d_x, K) + t$$

- From Camera Coord in O^2 to Pixel

$$\pi(R\pi^{-1}(x, d_x, K) + t, K)$$

- Get Pixel value in I_2

$$I_2\big(\pi(R\pi^{-1}(x, d_x, K) + t, K)\big)$$

$$\min_{R,t} \sum_x \left[I_2\big(\pi(R\pi^{-1}(x, d_x, K) + t, K)\big) - I_1(x)\right]^2$$

Tracking via Optimisation

Implementation Consideration:

- [R|t] parameterized with Lie-Group and Lie-Algebra
- Levenberg-Marquart algorithm
- Initial value of [R|t] must be close to true solution
- Computationally expensive for Jacobians:

$$J_F = \begin{bmatrix} \frac{\partial f_1}{\partial \omega_1} & \cdots & \frac{\partial f_1}{\partial \omega_6} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial \omega_1} & \cdots & \frac{\partial f_n}{\partial \omega_6} \end{bmatrix}, \text{ where}$$

$f_i = I_2\left(\pi\left(R\pi^{-1}(x^i, d_{x^i}, K) + t, K\right)\right) - I_1(x^i),\ i = 1, 2, \dots, n$ the number of pixels

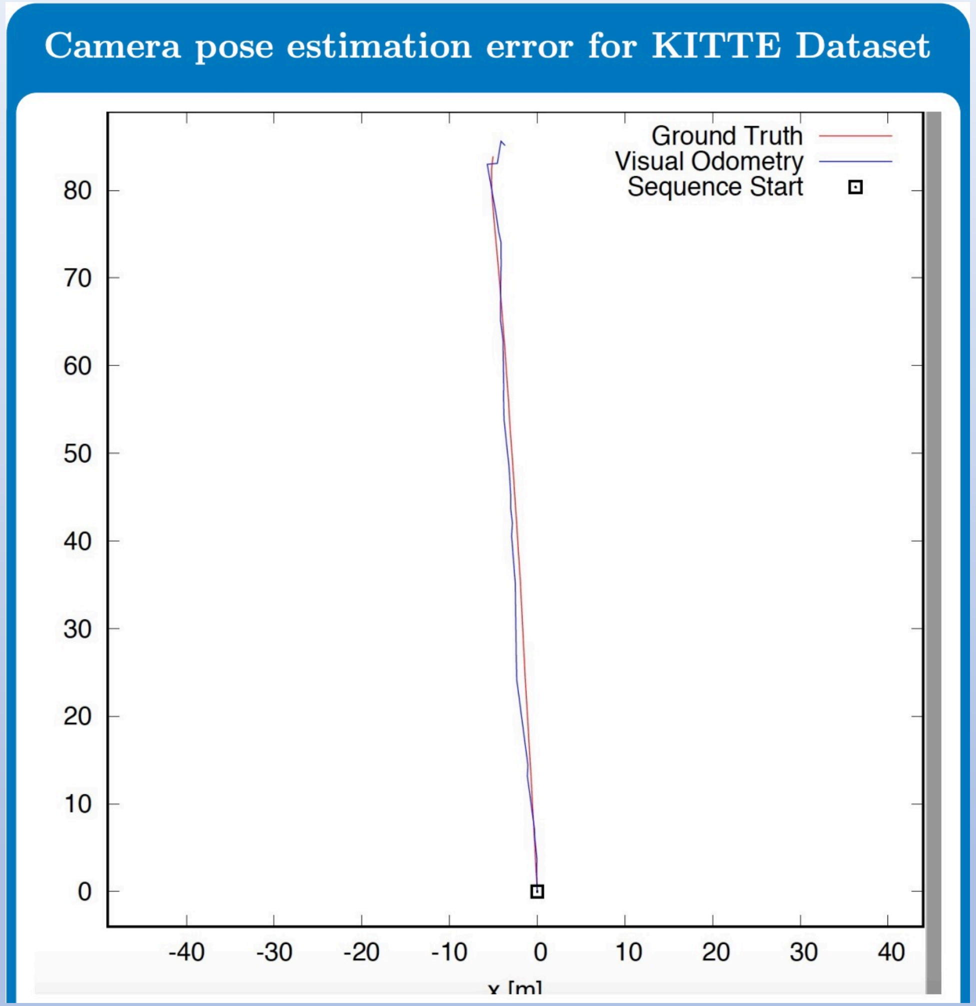
$\omega_i, i = 1,2,3,4,5,6$ *are parameterized* $[R|t]$ in lie-algebra

- Image pyramid for faster convergence
- 20-30ms/frame

Overview

- Introduction
- Stereo - from 2D to 3D
- Tracking via Optimisation
- Results
- Appendix

Results


Camera pose estimation error for KITTE Dataset
Ground Truth
Visual Odometry
Sequence Start
80
70
60
50
40
30
20
10
0
-40
-30
-20
-10
0
10
20
30
40
x [m]

Thank you!

Q & A

Appendix – Techincal Discussions

- Pin-hole camera model
- Disparity & depth
- Direct (photometric) v.s. In-direct(geometric, feature-based)
- Dense v.s. Semi-dense v.s. Sparse
- Non-linear Least Squares & Optimisation
- Code review: https://github.com/WangYuTum/odometry
- Deep Learning (Tracking, Segmentation)
- …

Appendix - Citations

[1]. Paul Furgale, Joern Rehder, Roland Siegwart (2013). Unified Temporal and Spatial Calibration for Multi-Sensor Systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan. (https://github.com/ethz-asl/kalibr)

[2]. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision, 47(1/2/3):7-42, April-June 2002. (http://vision.middlebury.edu/stereo/)

[3]. https://software.intel.com/sites/landingpage/IntrinsicsGuide/

[4]. https://opencv.org/

[5]. http://eigen.tuxfamily.org/index.php?title=Main_Page

[6]. https://strasdat.github.io/Sophus/