

# Stereo Camera Odometry and Environment Depth Mapping

Yulin Yang  
Mingchuan Zhao  
University of Delaware

# Outline

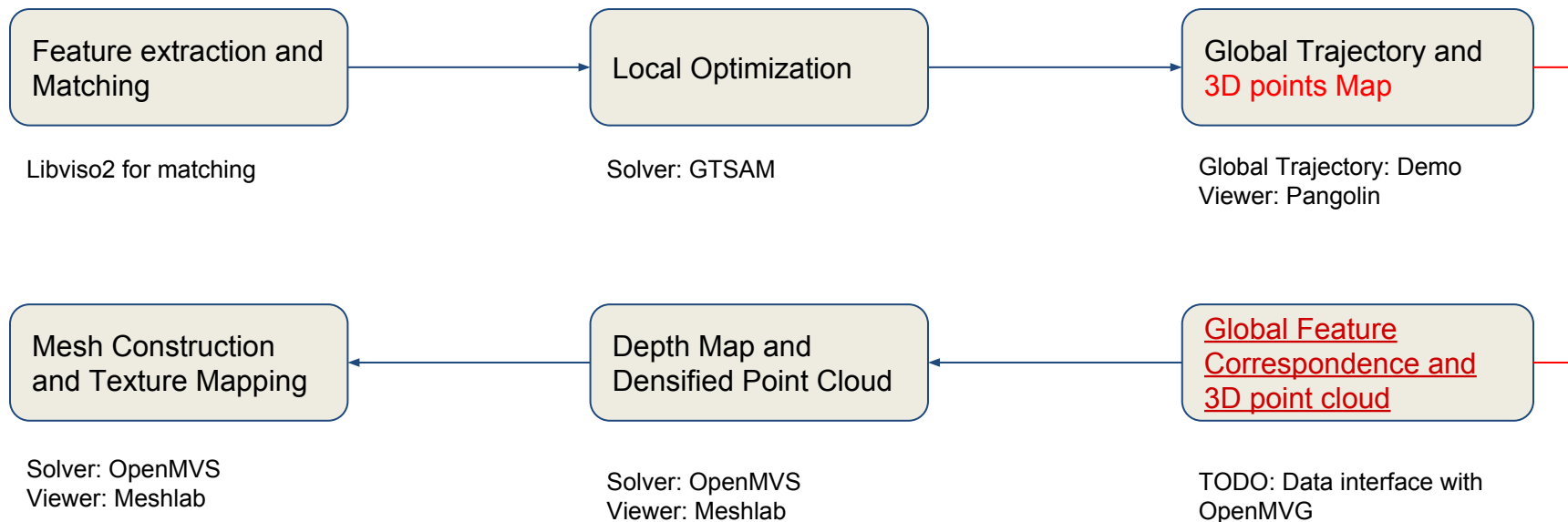
- ✓ Project Target
- ✓ Stereo Vision Model
- ✓ Pose Graph SLAM and Gtsam
- ✓ OpenMVG and OpenMVS
- ✓ Future Work

# Project Target

- Using Stereo Camera for 3D Reconstruction and Depth Mapping
- Be familiar with the following libraries:
  - \* libviso2[1]: A small efficient lib for stereo vision odometry
  - \* GTSAM[2]: an efficient lib for nonlinear least-squares optimization
  - \* Pangolin[3]: an lightweight lib for managing OpenGL displaying
  - \* OpenMVG[4]: an lib for Multi-View Structure from Motion
  - \* OpenMVS[5]: an lib for computing dense points cloud, surface etc.
  - \* KITTI Dataset[6]: A novel challenging real-world vision benchmarks suite.

# Project Target

## •The Major Processes

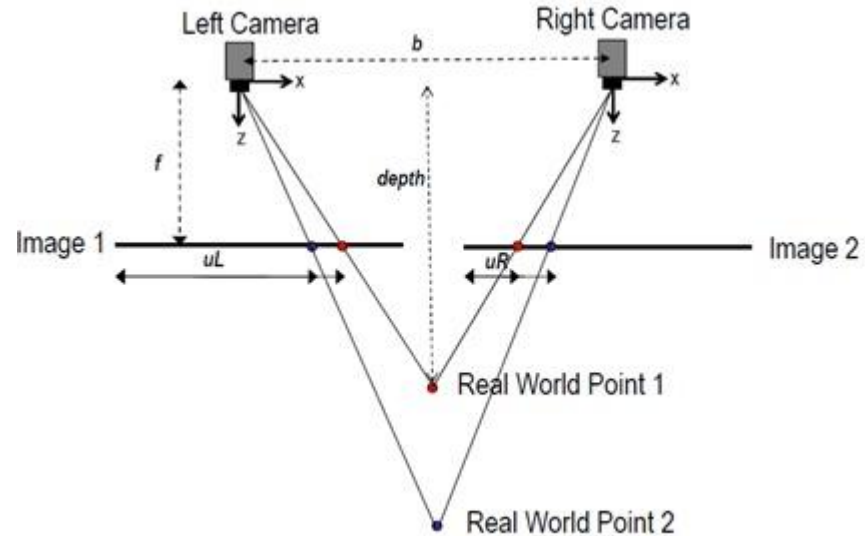


# Stereo Vision Model

- Stereo Vision Measurement Model

$$\begin{bmatrix} uR \\ uL \\ vL \end{bmatrix} = \begin{bmatrix} f_x * \frac{P_x}{P_z} + c_x \\ f_x * \frac{P_x - b}{P_z} + c_x \\ f_y * \frac{P_y}{P_z} + c_y \end{bmatrix}$$

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} (uR - c_x) \cdot \frac{b}{(uR - uL)} \\ (vL - c_y) \cdot \frac{f_x}{f_y} \cdot \frac{b}{uR - uL} \\ f_x \cdot \frac{b}{uR - uL} \end{bmatrix}$$



# Local Optimization With GTSAM

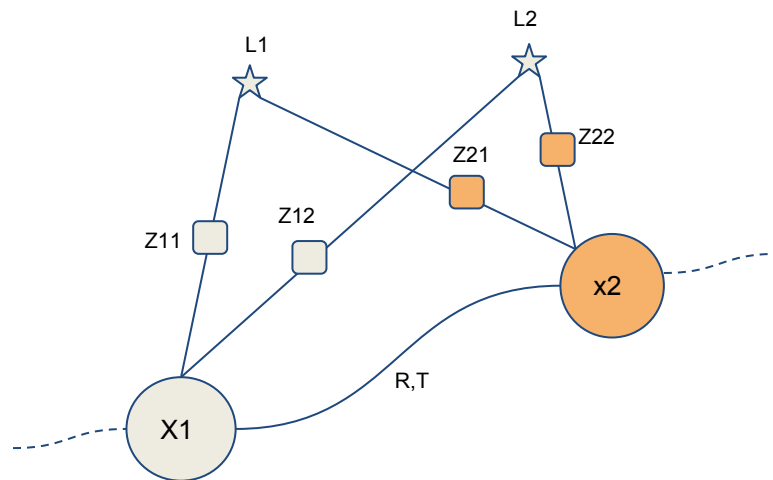
- Local Bundle Adjustment with GTSAM
- (GTSAM is a popular solver for weighted least squares)

The measurement function:

$$z_{ij} = h(x_i, L_j)$$

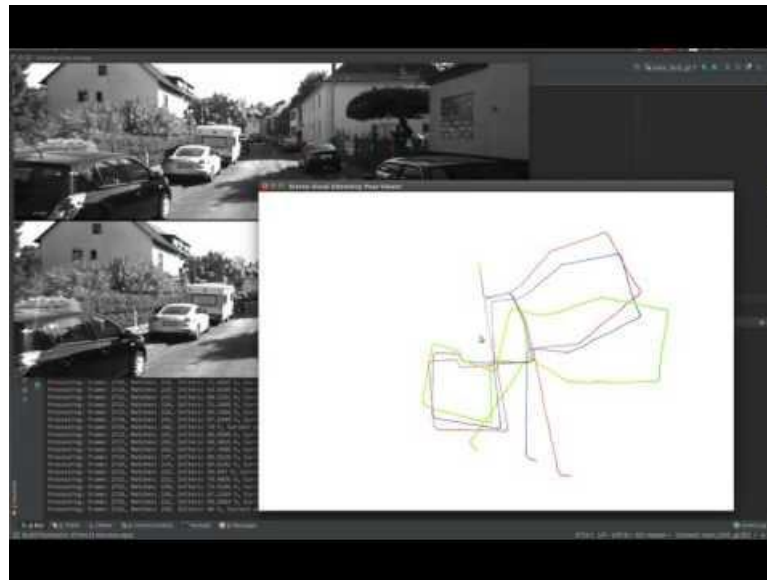
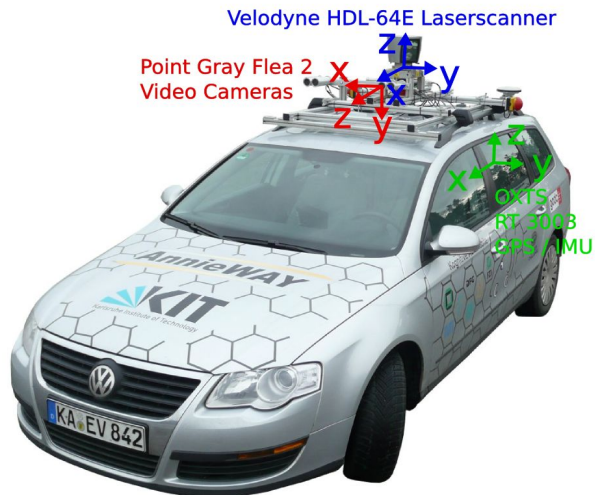
The cost function:

$$\sum_i \sum_j \|z_{ij} - h(x_i, L_j)\|_{(\Sigma_{ij})^{-1}}^2$$



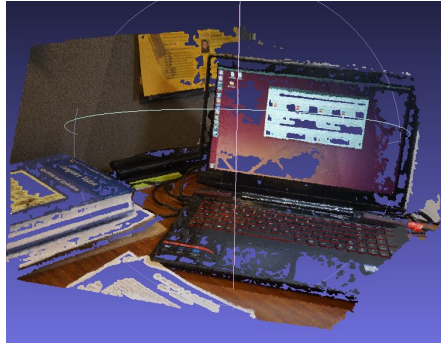
# Stereo Vision Odometry Demo

## System Setup For KITTI Dataset



- Complete Video : <https://www.youtube.com/watch?v=QbXD9BVGacc>
- Source Code: <https://github.com/yangyulin/SVO.git>

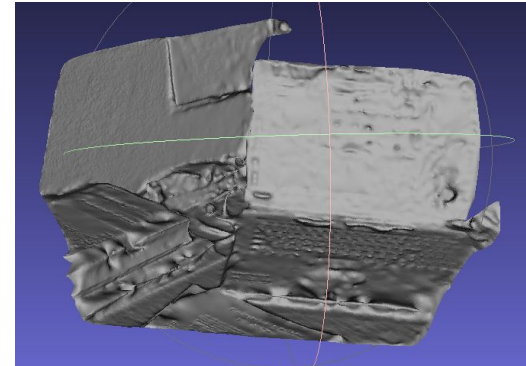
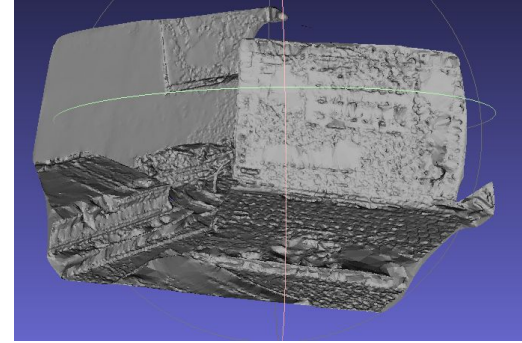
# OpenMVG and OpenMVS



Densify Point  
Cloud



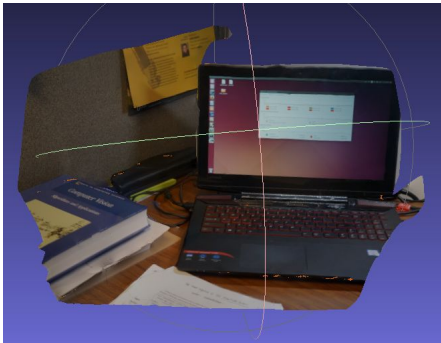
Reconstruct 3D  
Mesh



3D Texture



Refine 3D  
Mesh

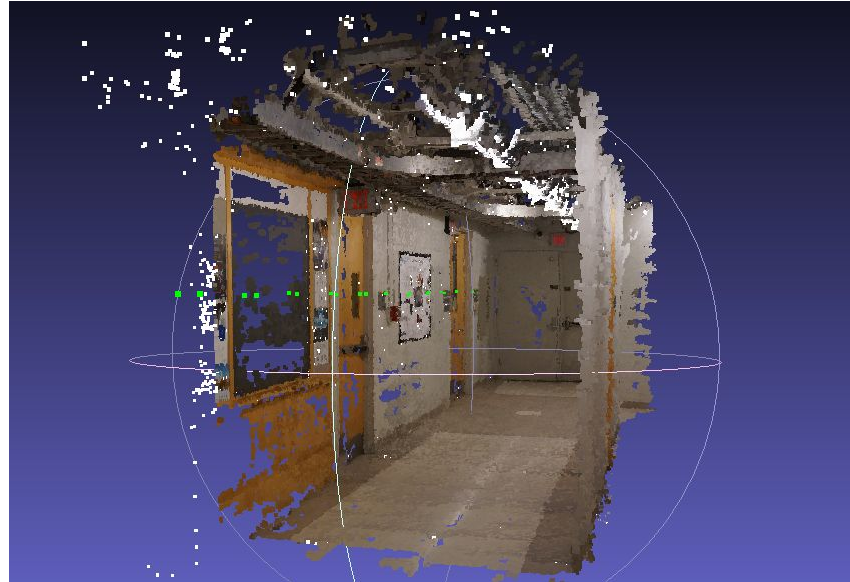




# OpenMVG and OpenMVS Demo

- General Procedures for using OpenMVG

1. Image listing
2. Image description computation
3. Corresponding images and correspondences computation
4. SfM solving
5. OpenMVS for post processing
6. Meshlab for displaying



# Reference List

- [1] A. Geiger, J. Ziegler, and C. Stiller, “Stereoscan: Dense 3d reconstruction in real-time,” in *Intelligent Vehicles Symposium (IV)*, 2011.
- [2] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” 2012.
- [3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: a versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics*, vol. abs/1502.00956, 2015.
- [4] P. Moulon, P. Monasse, and R. Marlet, “Openmvg (open multiple view geometry),” 2012.
- [5] P. Moulon, “Openmvs (open multiple-view stereo),” 2012.
- [6] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

# Thanks a lot!

Q&A