NetCalib: A Novel Approach for LiDAR-Camera Auto-Calibration based on Deep Learning

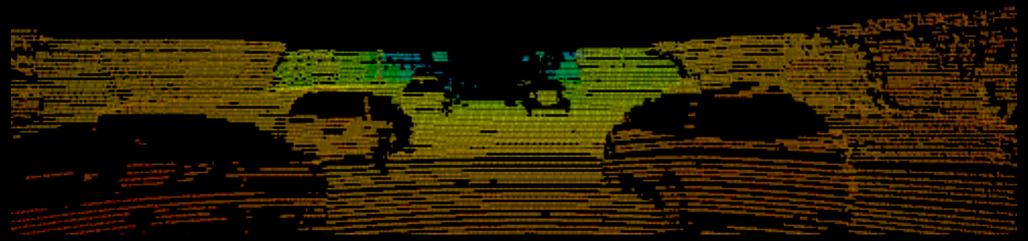
Shan Wu, Amnir Hadachi, Damien Vivet, Yadu Prabhakar

ITS Lab
Institute of Computer Science
University of Tartu
Shan.Wu@ut.ee

The 25th International Conference on Pattern Recognition (ICPR2020) - MiCo Milano Congress Center, ITALY 10 - 15 January 2021

Introduction

Camera



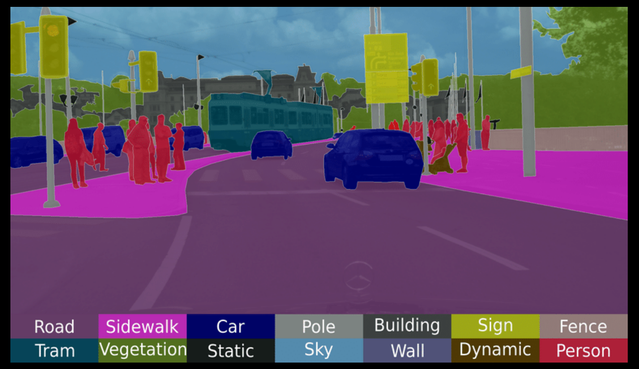
LiDAR

Introduction

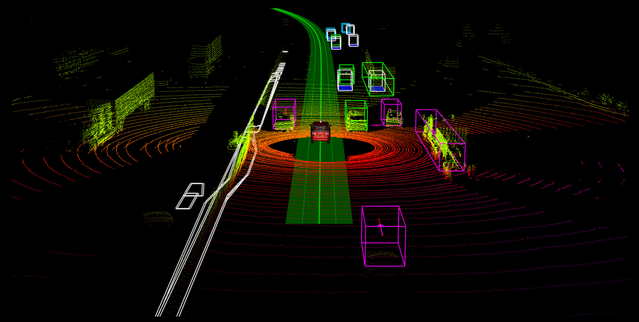
Fusion

Introduction

Fusion



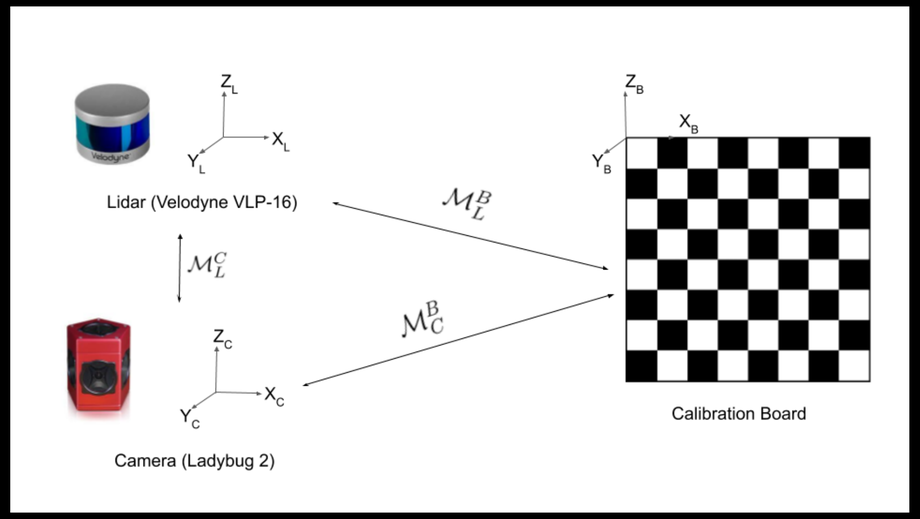
Semantic Segmentation



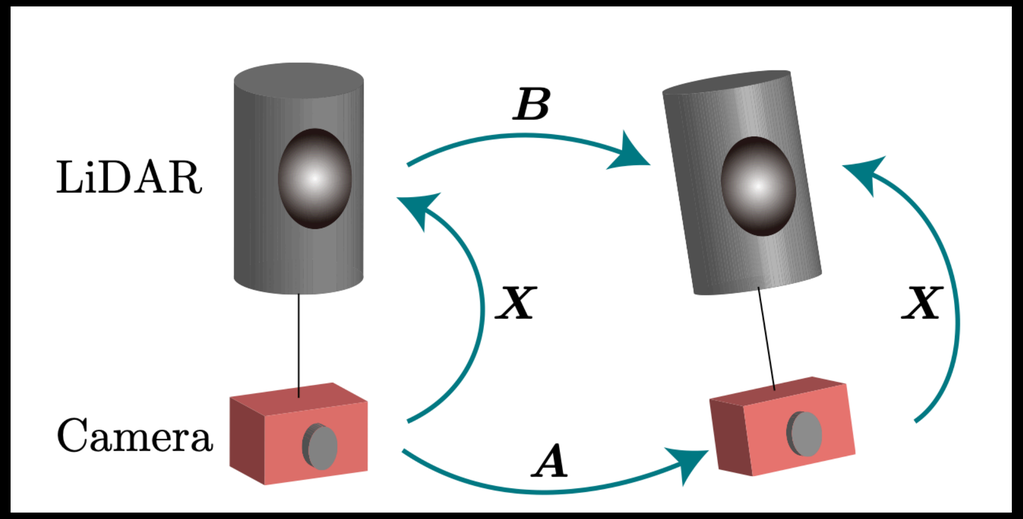
Object Detection

Scene Reconstruction

Literature

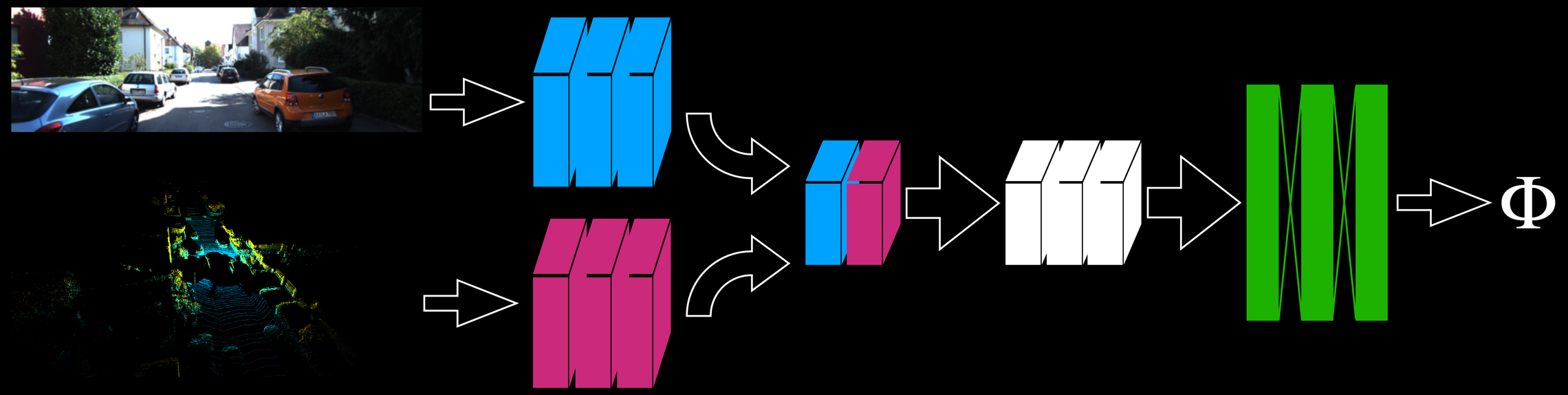


Target-based



Motion-based

Literature



ML-based

Contributions

- Open sourced on GitHub
- New ML model
- Uses depth maps for inputs instead of disparities or RGB images

https://github.com/simonwu53/NetCalib-Lidar-Camera-Auto-calibration

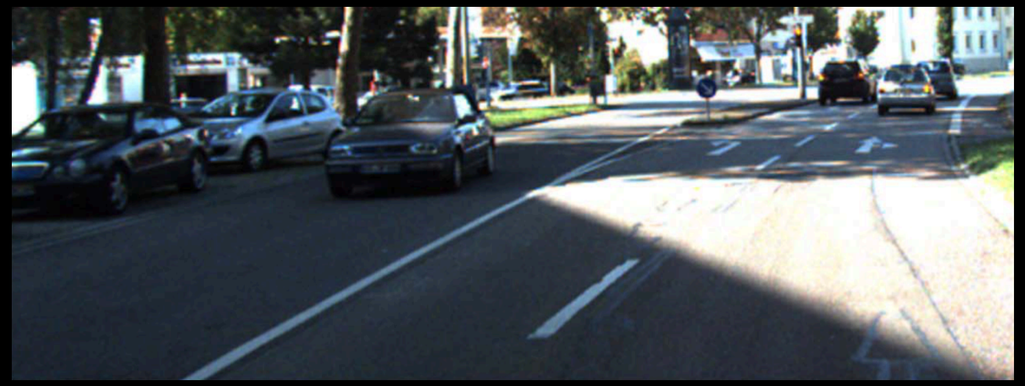
Methodology

- Stereo Matching
- LiDAR Projection
- Artificial Error Generation
- Deep Neural Network

Stereo Matching

- Semi Global Block Matching
- ML methods based on RGB images

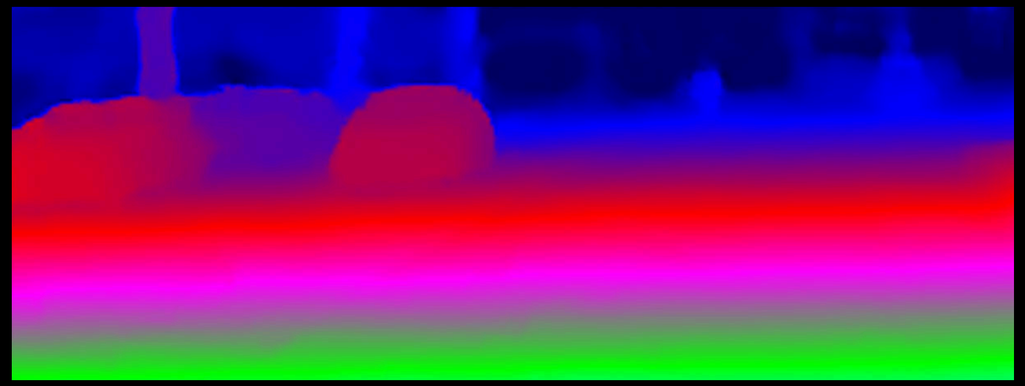
$$depth = \frac{B \cdot f}{disparity}$$



Stereo Matching

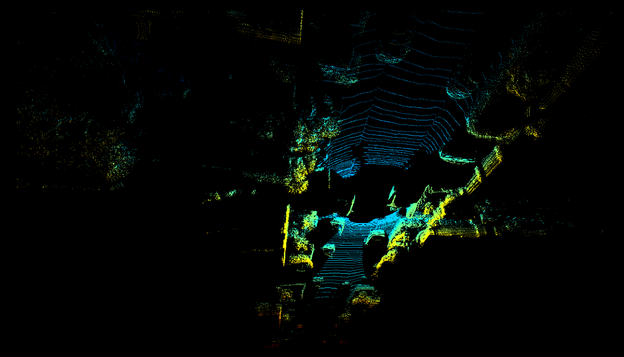
- Semi Global Block Matching
- ML methods based on RGB images

$$depth = \frac{B \cdot f}{disparity}$$



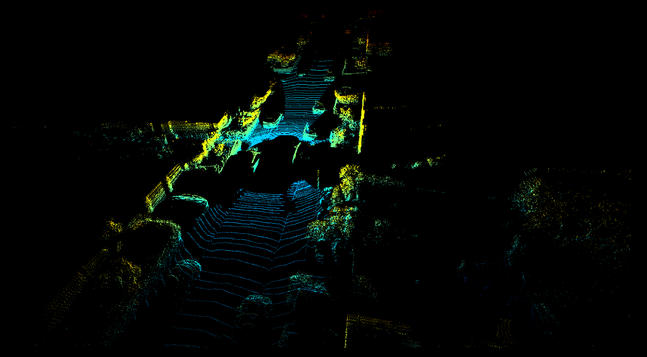
LiDAR Projection

$$T = \begin{bmatrix} R(r_x, r_y, r_z) & [t_x, t_y, t_z]^T \\ 0 & 1 \end{bmatrix}$$



LiDAR Projection

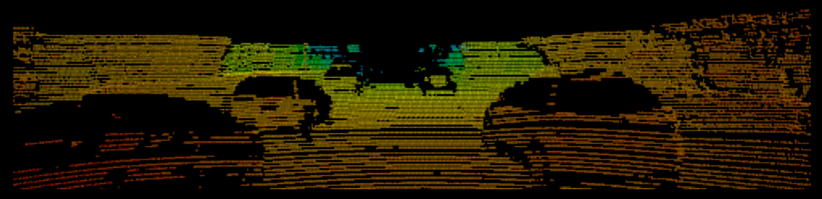
$$T = \begin{bmatrix} R(r_x, r_y, r_z) & [t_x, t_y, t_z]^T \\ 0 & 1 \end{bmatrix}$$



LiDAR Projection

$$T = \begin{bmatrix} R(r_x, r_y, r_z) & [t_x, t_y, t_z]^T \\ 0 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$$[u, v, 1]^T = PT[x, y, z, 1]^T$$

Error Generation

- Rotation range: +/- 2 degrees
- Translation range: +/- 20 cm
- Uniform distribution
- Random selected

Error Generation

- Rotation range: +/- 2 degrees
- Translation range: +/- 20 cm
- Uniform distribution
- Random selected

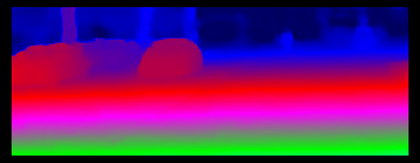
$$e = [r_x, r_y, r_z, a, b, c]$$

$$T_e = \begin{bmatrix} R(r_x, r_y, r_z) & [a, b, c]^T \\ 0 & 1 \end{bmatrix}$$

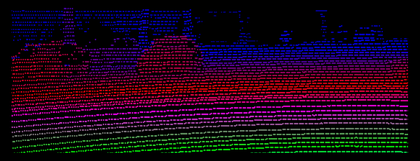
$$[u', v', 1]^T = PT_eT[x, y, z, 1]^T$$

$$[u, v, 1]^T = PT^{-1}T_eT[x, y, z, 1]^T$$

Deep Neural Network

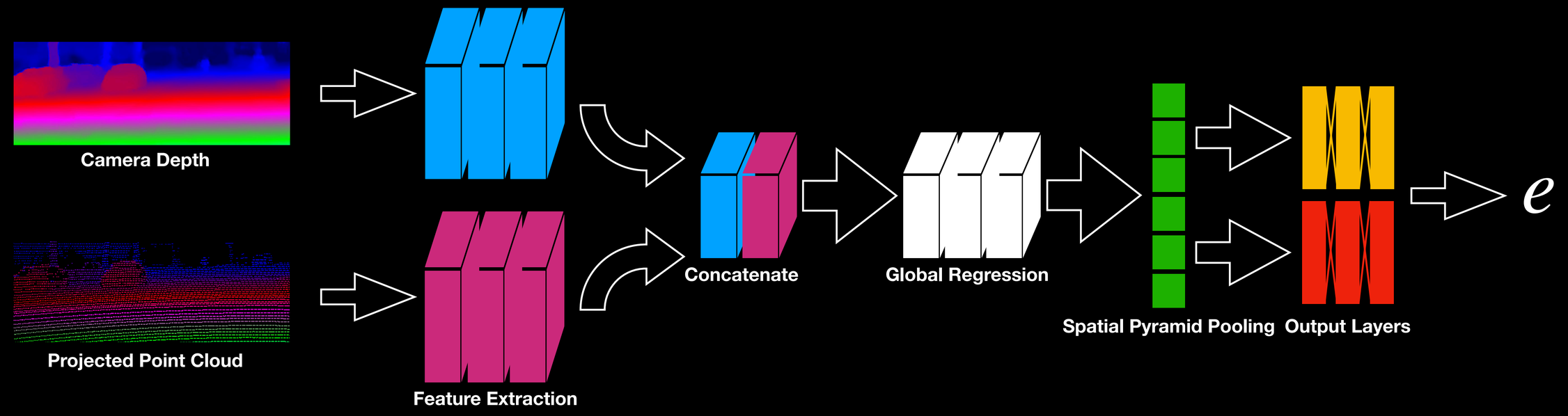


Camera Depth

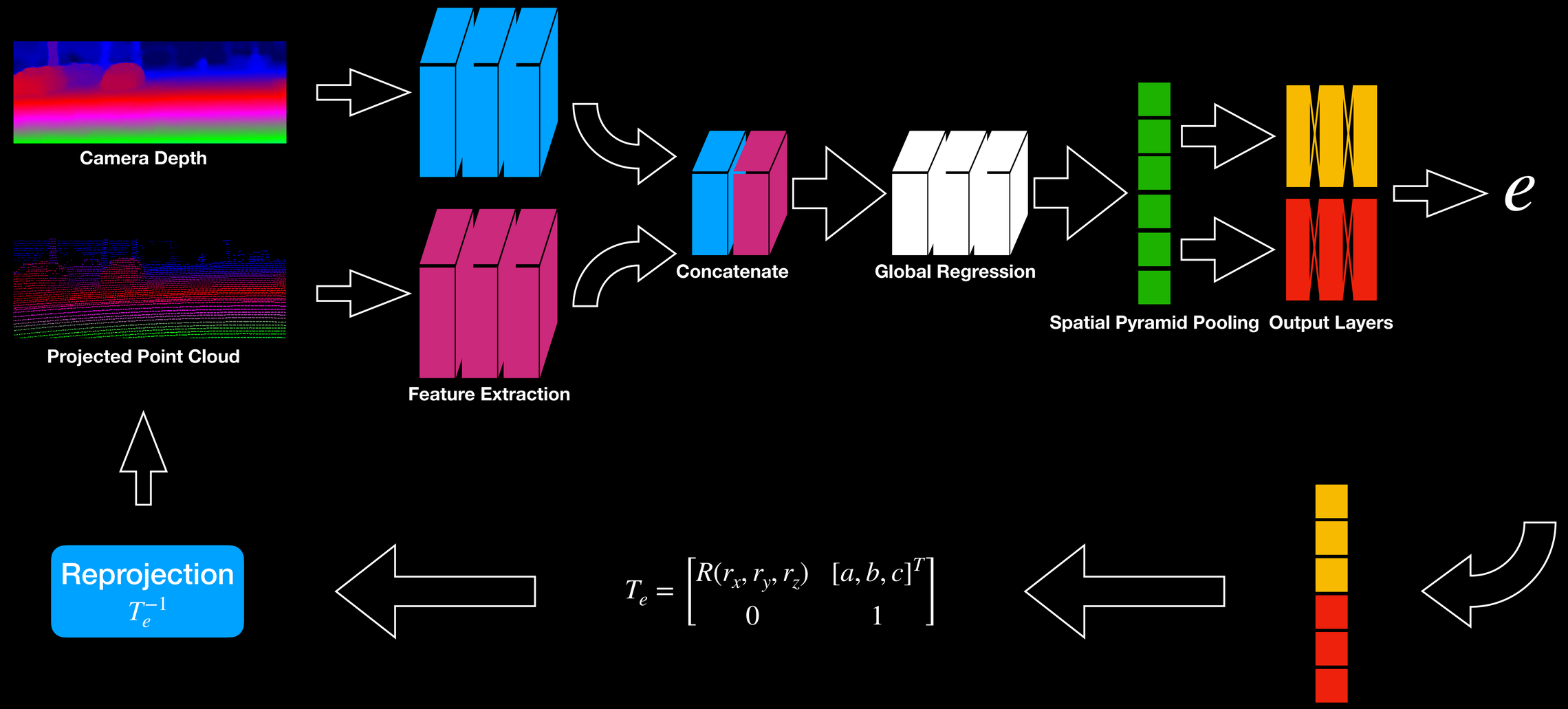


Projected Point Cloud

Deep Neural Network



Deep Neural Network



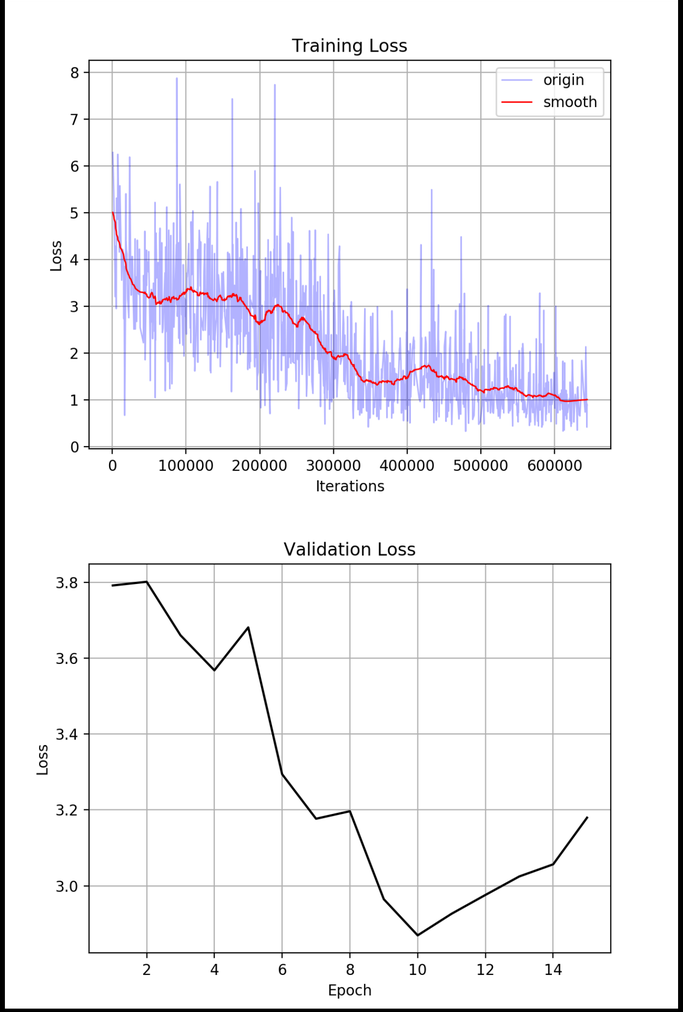
Experiments

- L1 loss
- KITTI dataset
- Rotation: -0.04°, 0.16°, 0.09°
- Translation: 1.20 cm, 2.77 cm, -1.10 cm
- Training speed: 0.055s/it
- Testing speed: 0.016s/it

Dataset	Number of Frames
Training	42949
Validation	3168
Testing	258

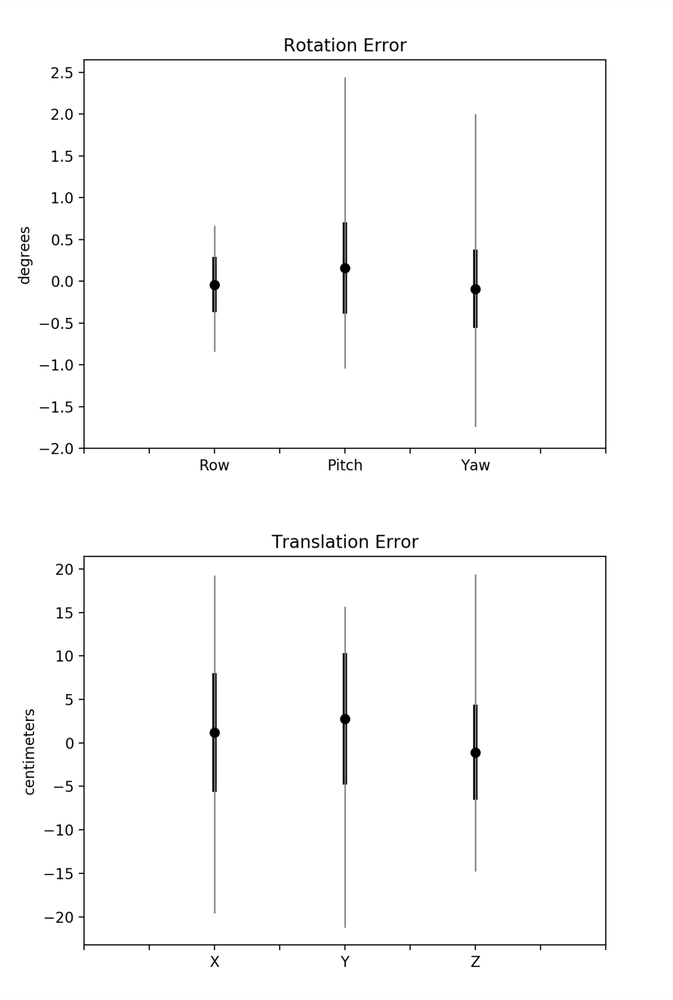
Experiments

- L1 loss
- KITTI dataset
- Rotation: -0.04°, 0.16°, 0.09°
- Translation: 1.20 cm, 2.77 cm, -1.10 cm
- Training speed: 0.055s/it
- Testing speed: 0.016s/it



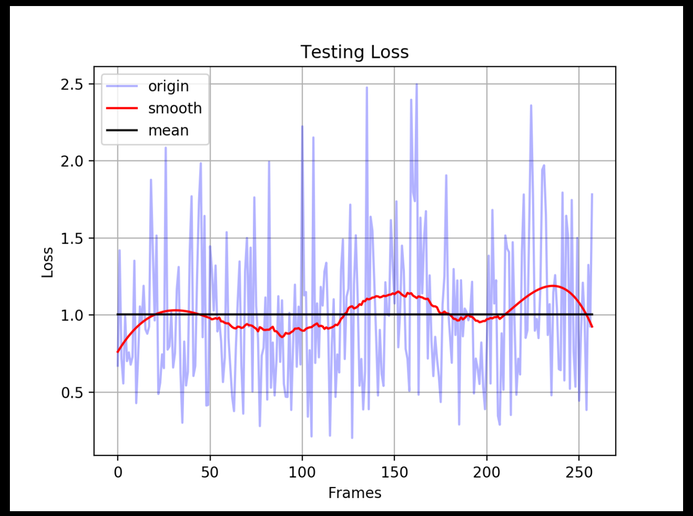
Experiments

- L1 loss
- KITTI dataset
- Rotation: -0.04°, 0.16°, 0.09°
- Translation: 1.20 cm, 2.77 cm, -1.10 cm
- Training speed: 0.055s/it
- Testing speed: 0.016s/it



Experiments

- L1 loss
- KITTI dataset
- Rotation: -0.04°, 0.16°, 0.09°
- Translation: 1.20 cm, 2.77 cm, -1.10 cm
- Training speed: 0.055s/it
- Testing speed: 0.016s/it



Visualization

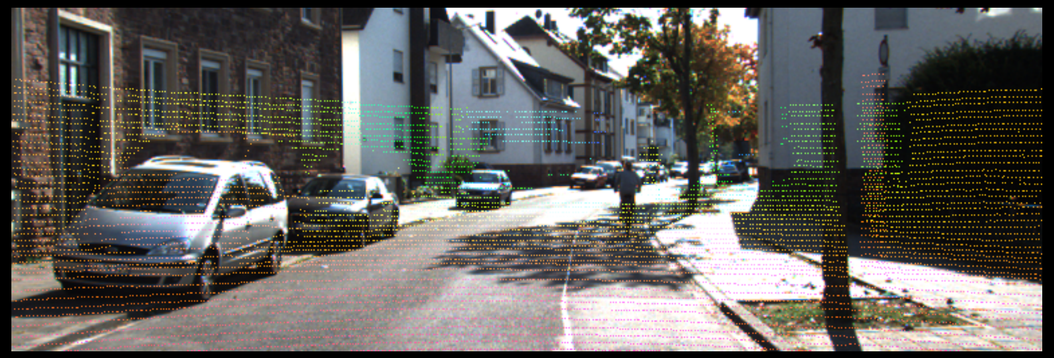
Projection with errors

Camera depth

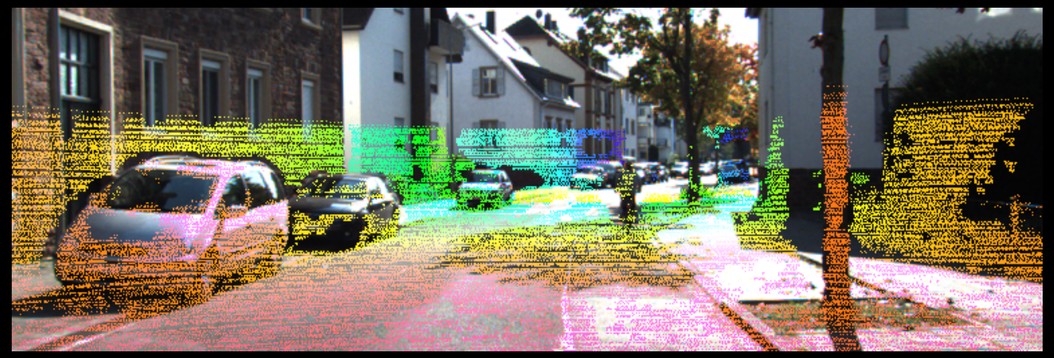
Projection with model inference

Projection without errors

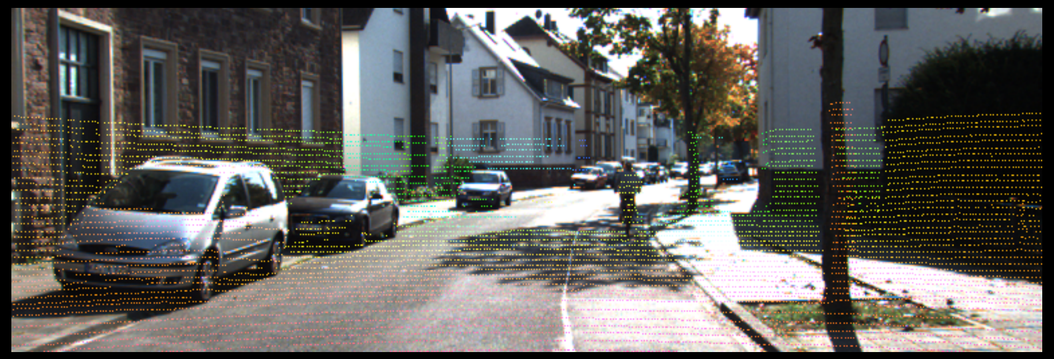
Visualization



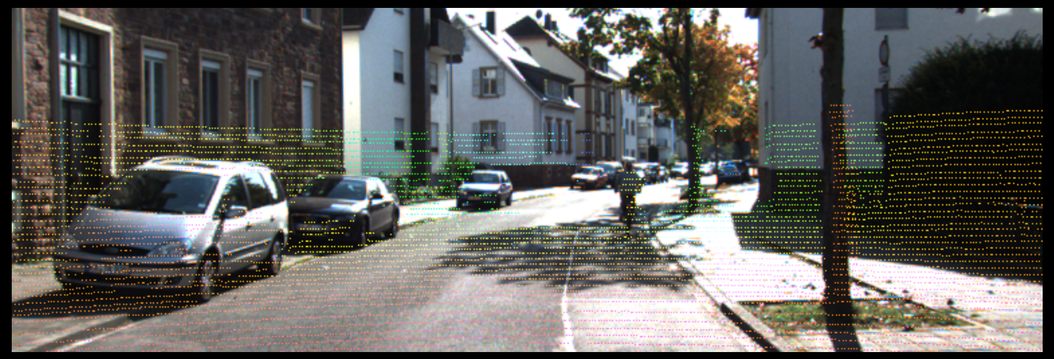
Projection with errors



Camera depth



Projection with model inference



Projection without errors

Conclusions

- Proposed methodology can be used for sensor auto-calibration
- Model works with arbitrary input dimensions
- Open sourced
- Future perspectives:
    - More error ranges
    - Fine-tune depth estimation for stereo cameras
    - LiDAR depth interpolation

Thanks for listening

E-mail : shan.Wu@ut.ee
GitHub: https://github.com/simonwu53/NetCalib-Lidar-Camera-Auto-calibration