



Long Range Depth Sensing

Shuhang Liao

December 10, 2020

Background

Existing depth sensing methods

Methods	Range	Dependency
Projection	50m+	cam1/3/4 pose/intrinsics
Stereo	50m+	cam4/17 intrinsics/extrinsics
MonoDepth	50m+	cam4 intrinsics
Tele	50~400m	\
Lidar Fusion	0~200m	\
Radar	0~150m	\

Problem and Solution

There are several challenges in traditional stereo camera-based solutions

1. The baseline/depth ratio is very small, thus triangulation is usually ill-conditioned.
2. Difficulty in calibrating small-FOV cameras.
3. Maintaining the calibration during usage.

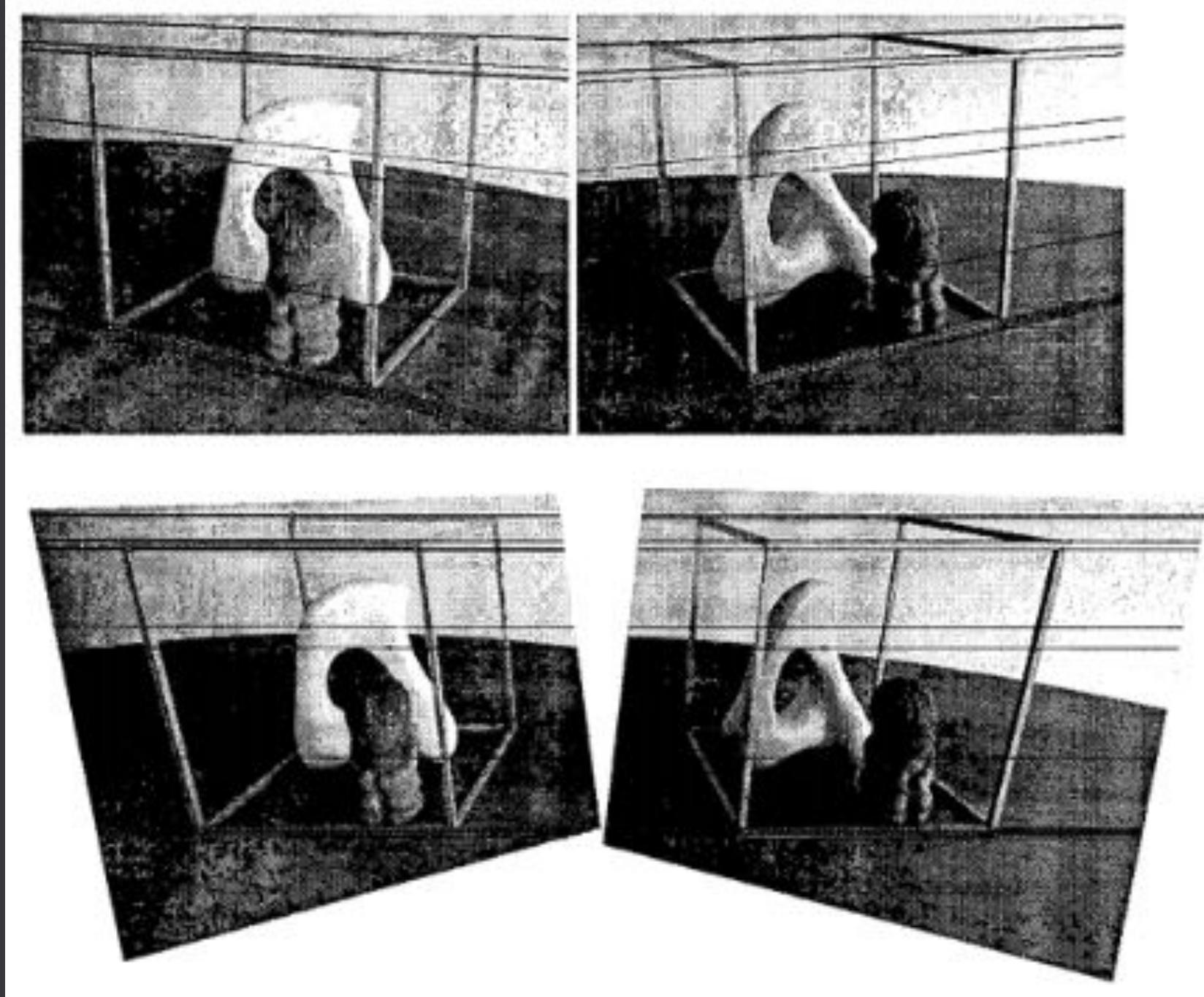
New solution:

Depth Sensing Beyond LiDAR Range in CVPR 2020.

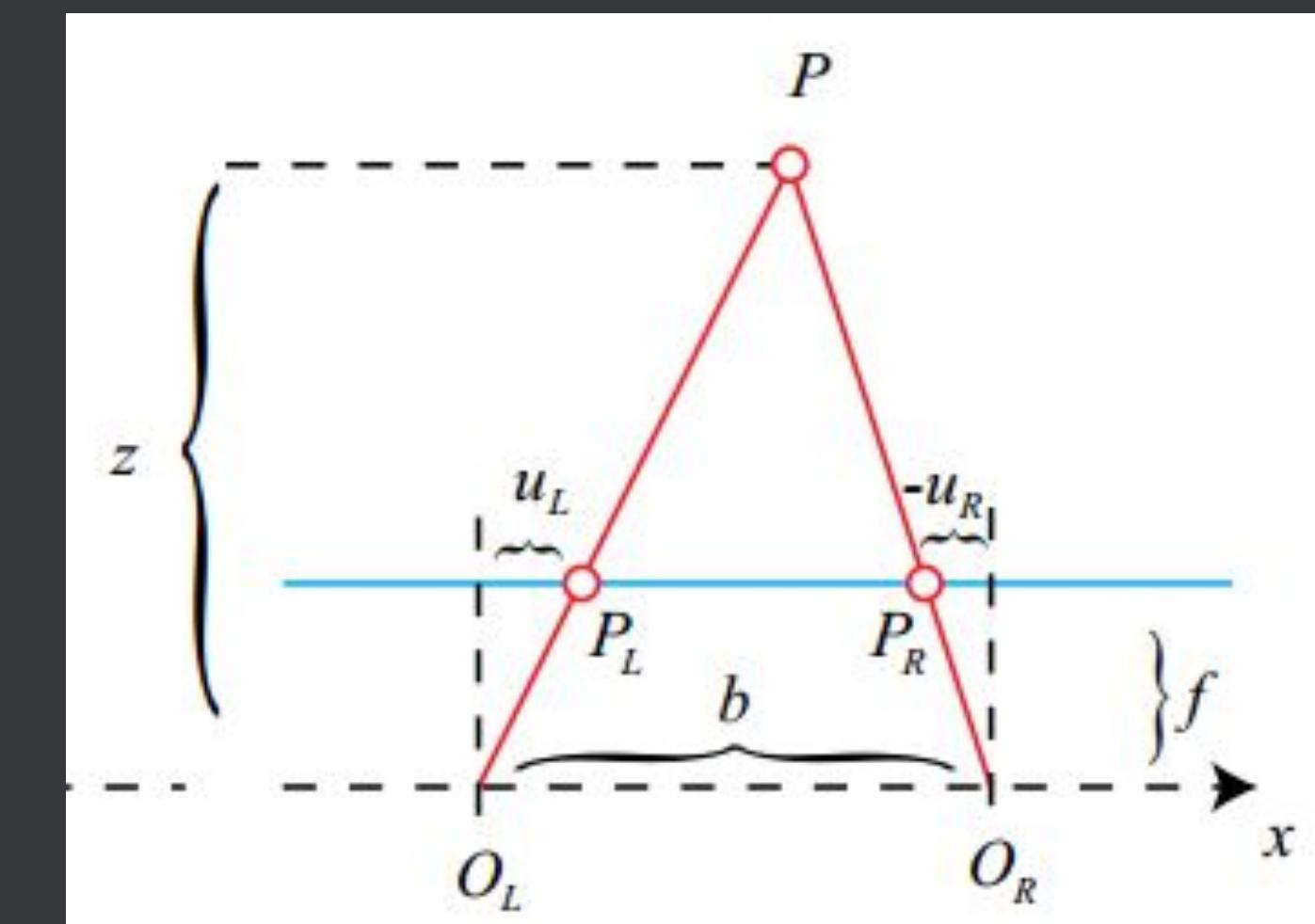
Features:

1. Online stereo camera pseudo rectification.
2. Depth sensing requires only partial camera calibration.
3. Need a specific third camera.

Depth Sensing Beyond LiDAR Range in CVPR 2020

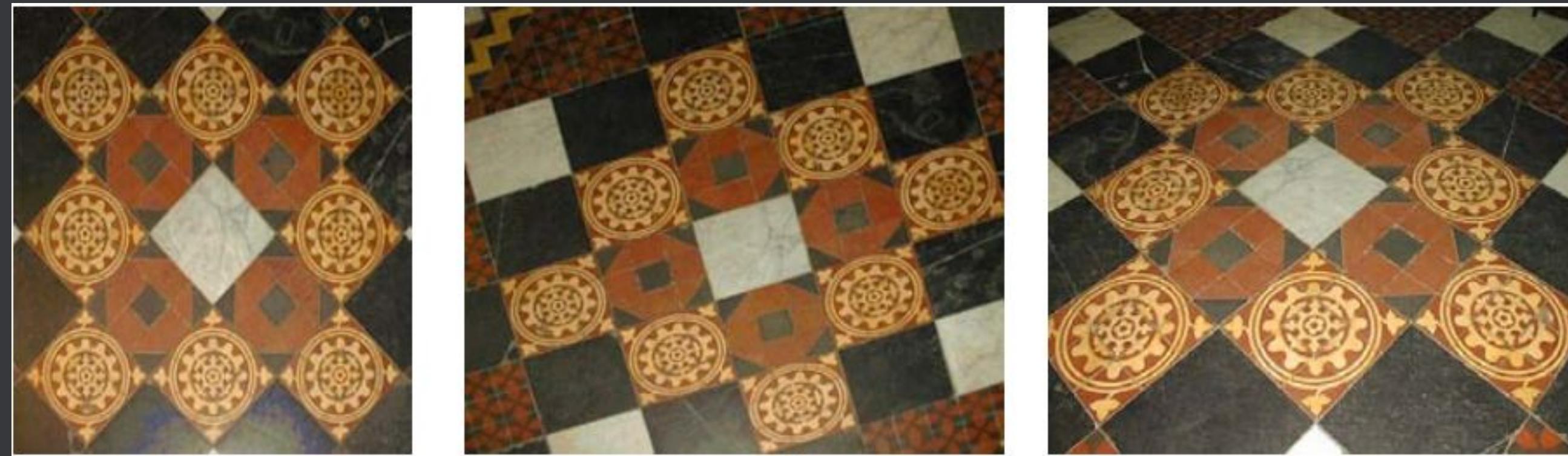


stereo rectification



def of disparity

Depth Sensing Beyond LiDAR Range in CVPR 2020

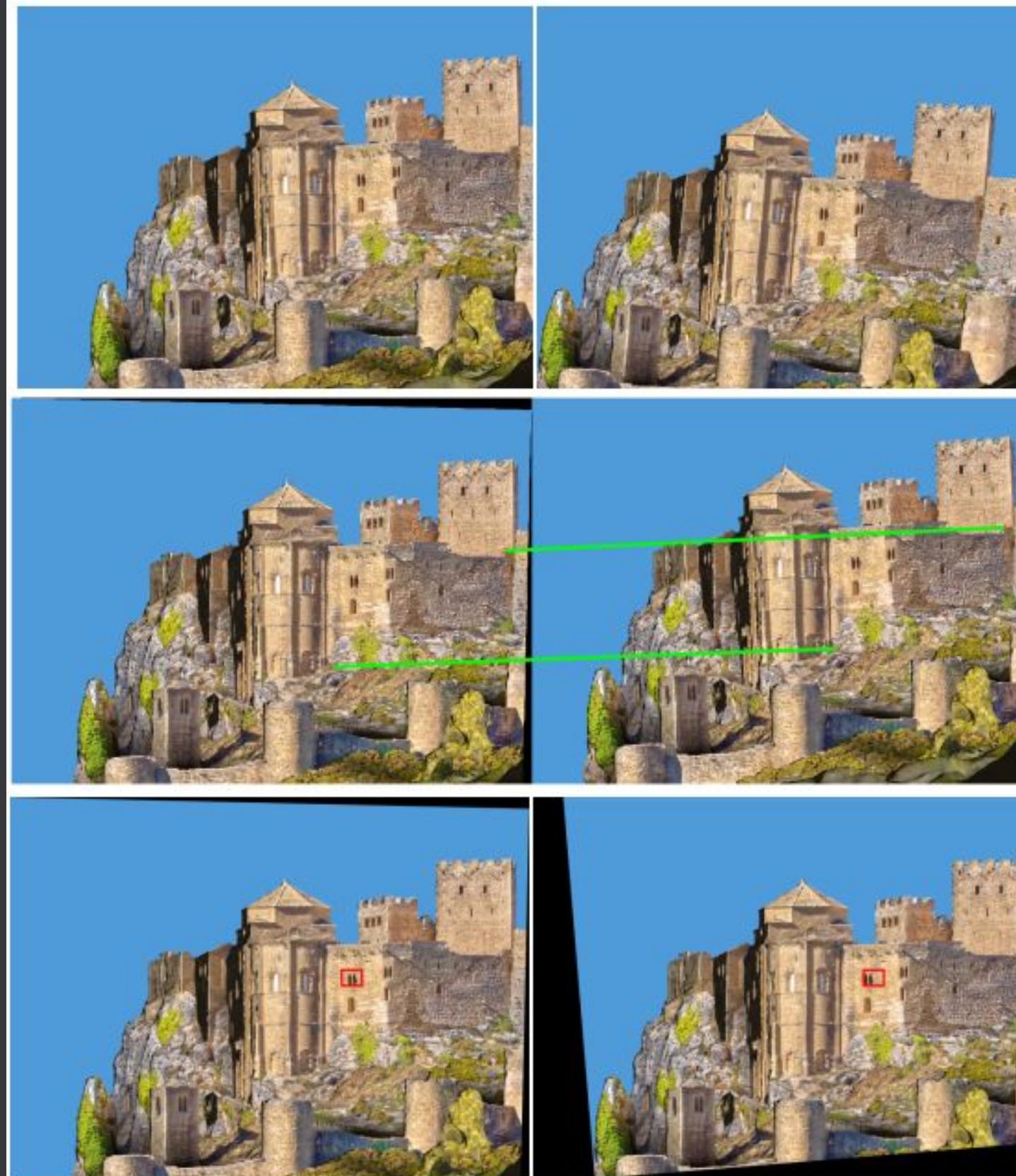


Similarity & Affine & Projective

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

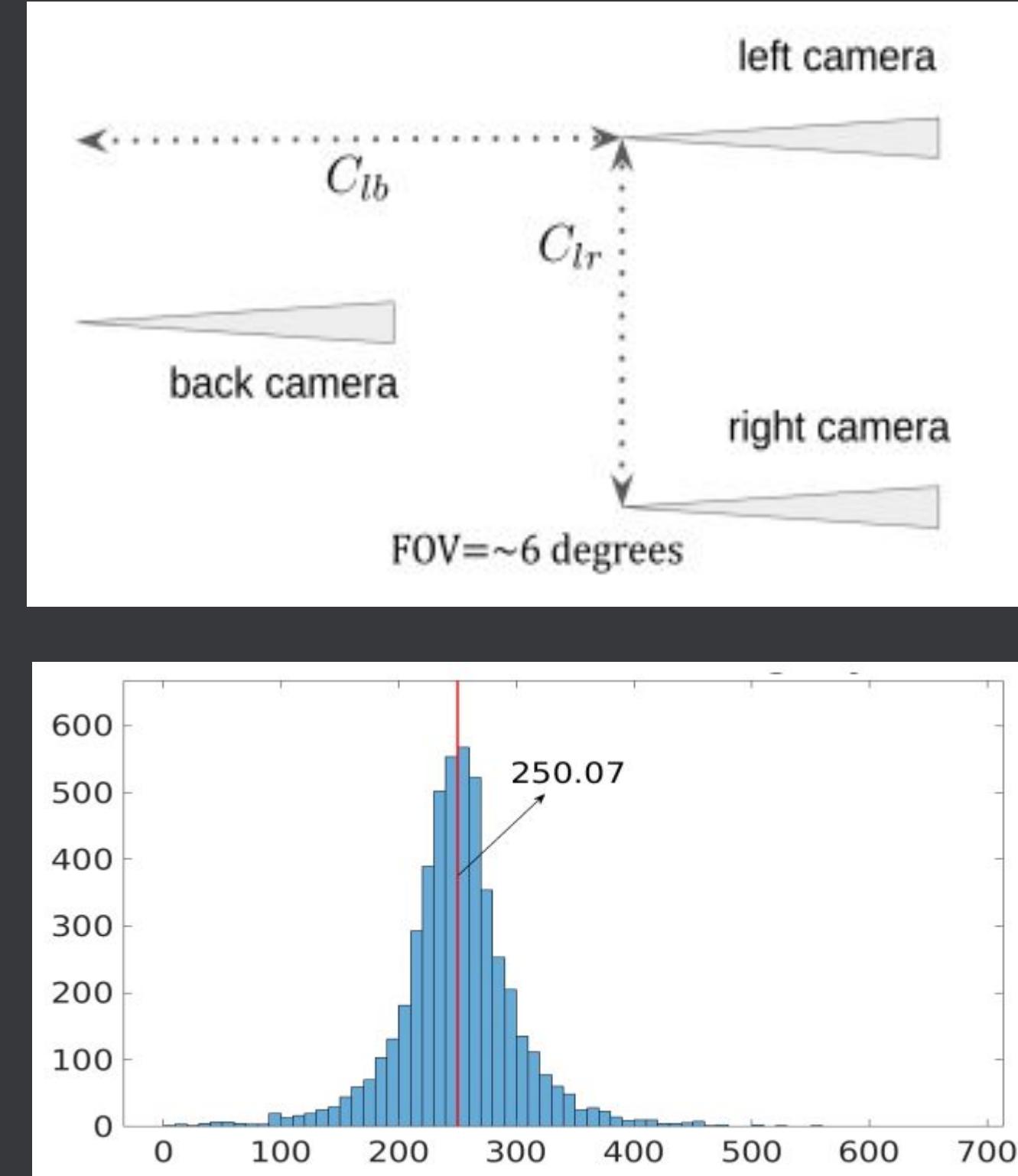
Affine Transformation

Depth Sensing Beyond LiDAR Range in CVPR 2020



pseudo rectification

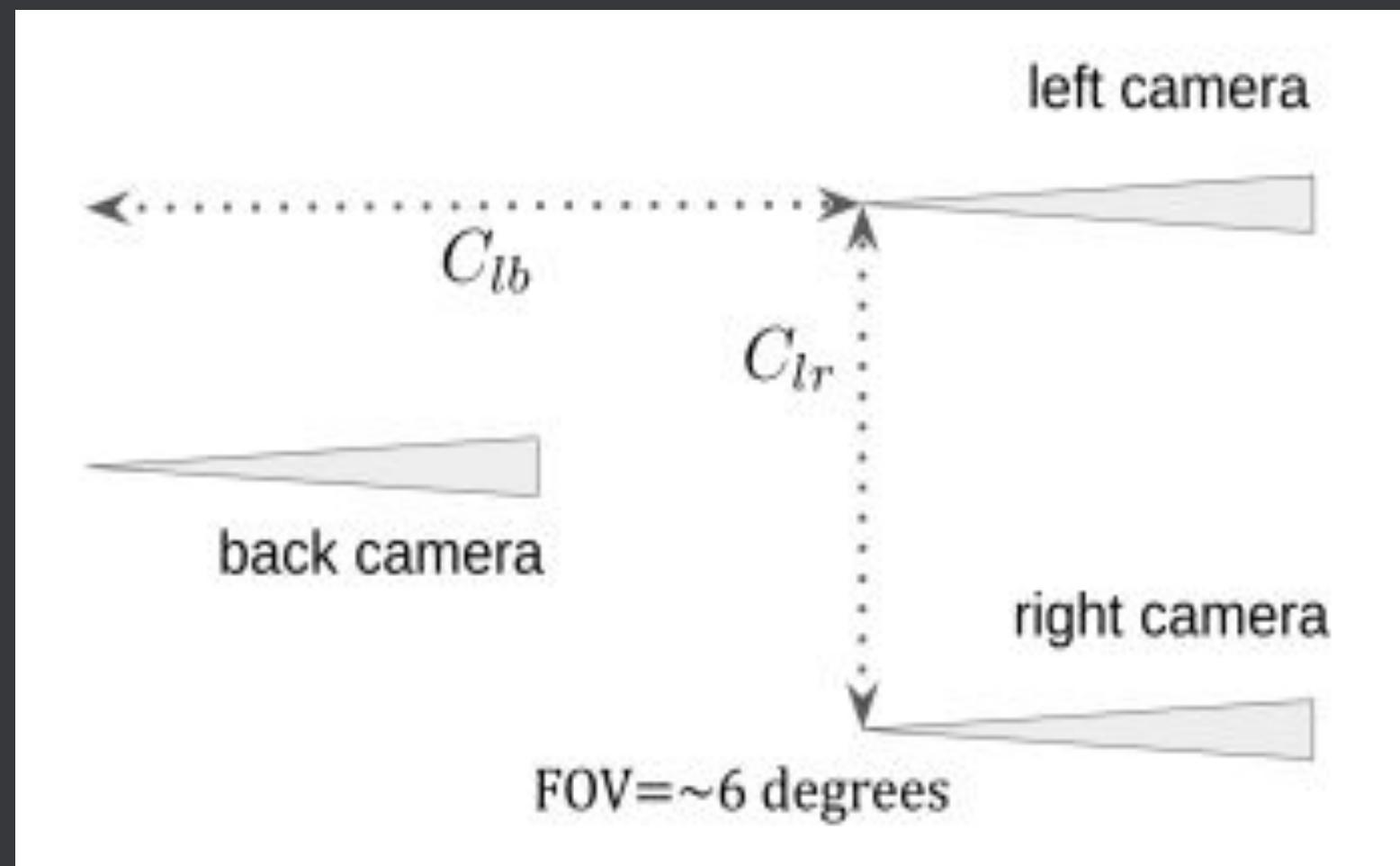
disparity estimation



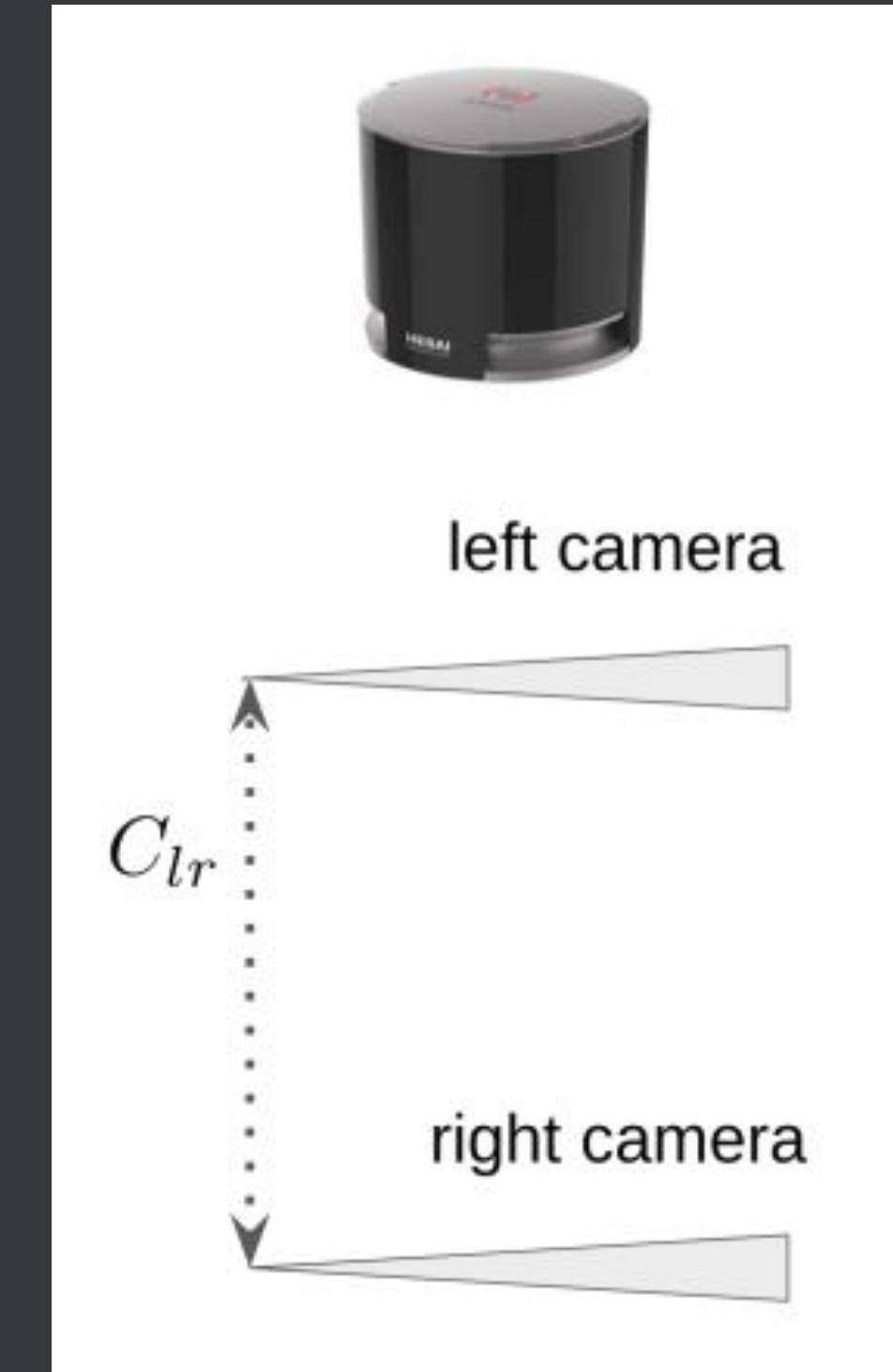
global disparity bias removal

Lidar-based Implementation

0 Hardware



Paper



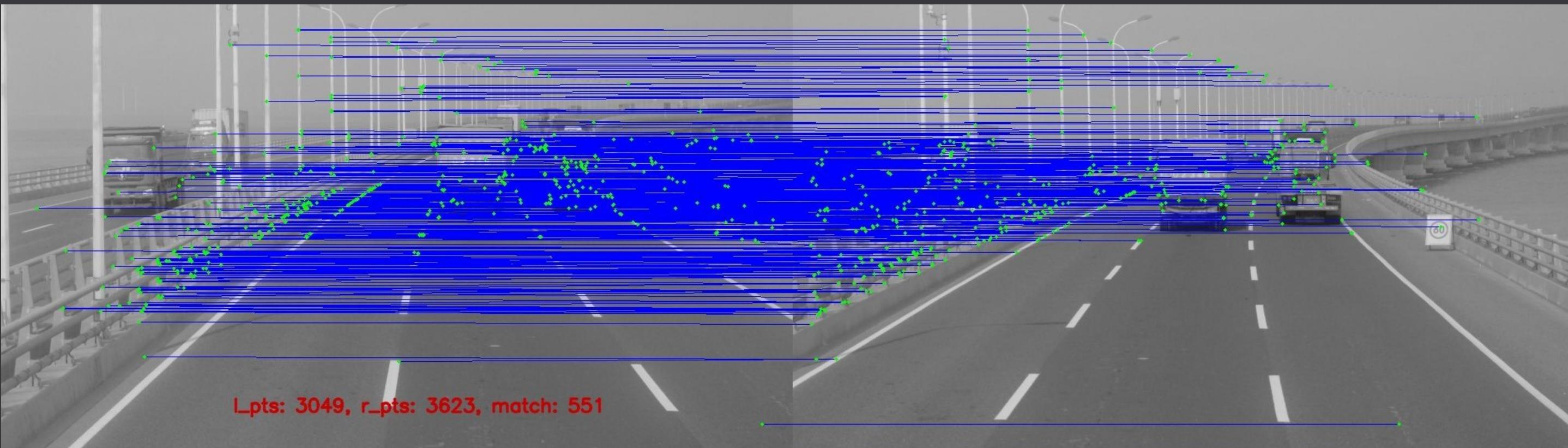
ours

Lidar-based Implementation

1 Feature detecting and matching



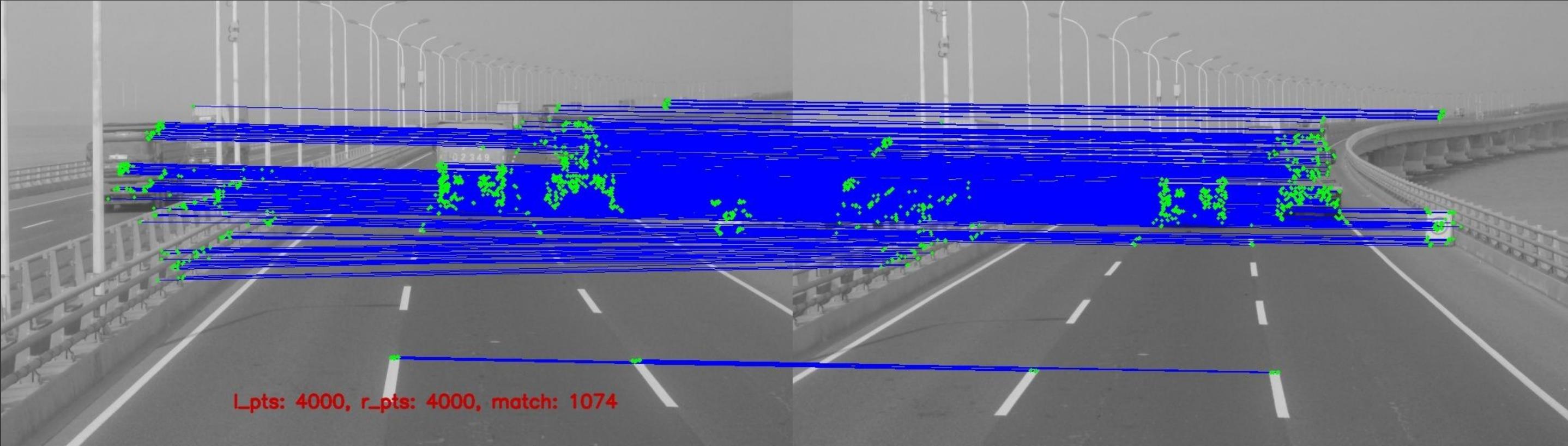
Original



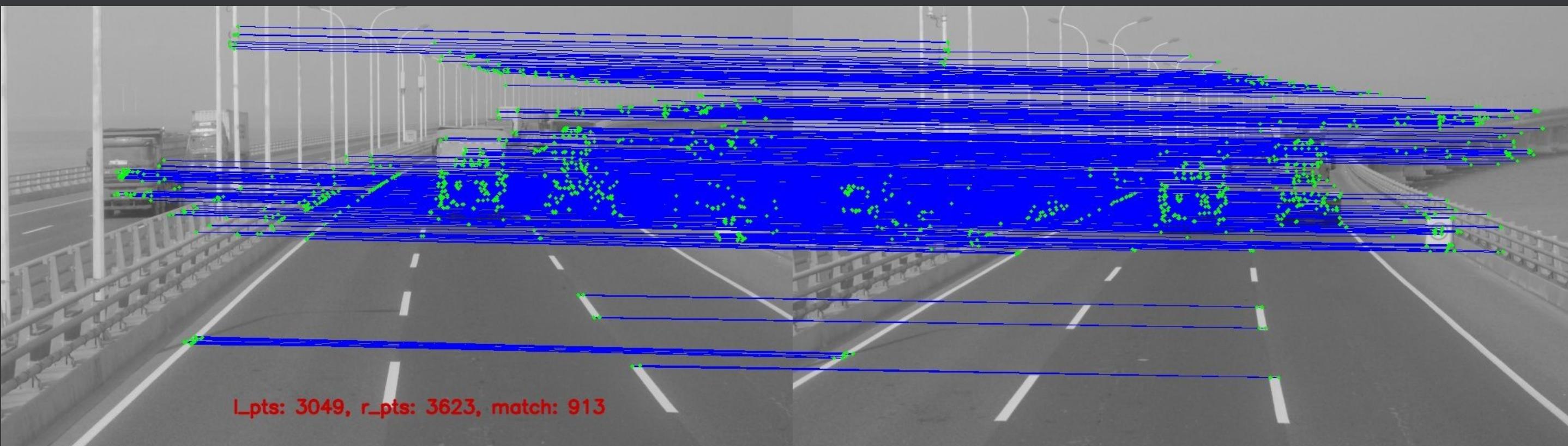
Traditional stereo matching

Lidar-based Implementation

1 Feature detecting and matching



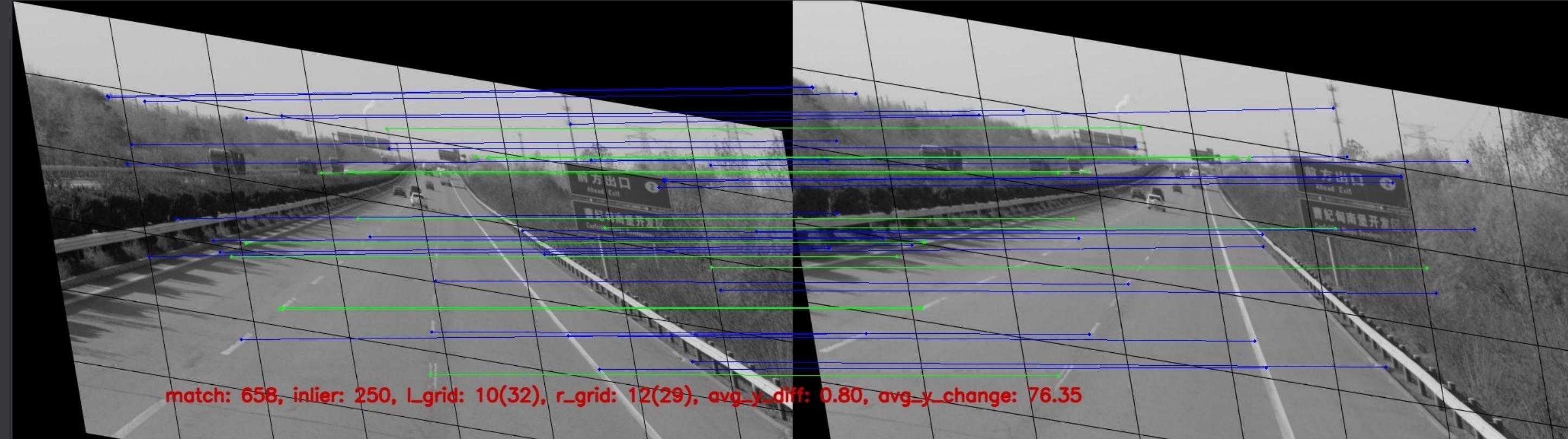
detecting with OpenCV, brutal matching, GMS filter



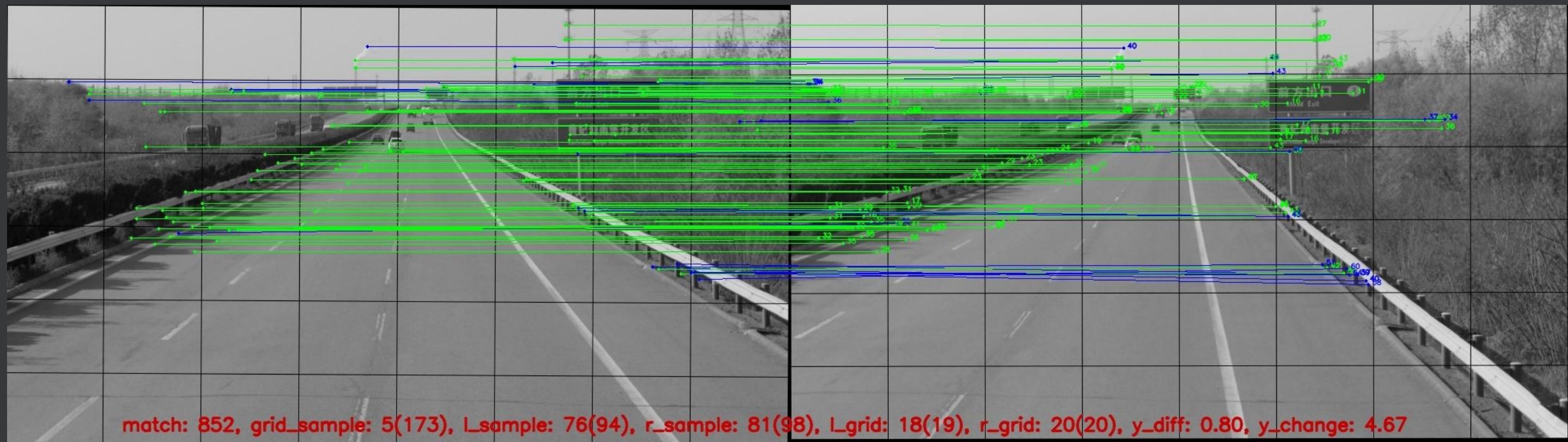
detecting with ORBSLAM, brutal matching, GMS filter

Lidar-based Implementation

2 Pseudo Rectification



RANSAC + SVD in paper



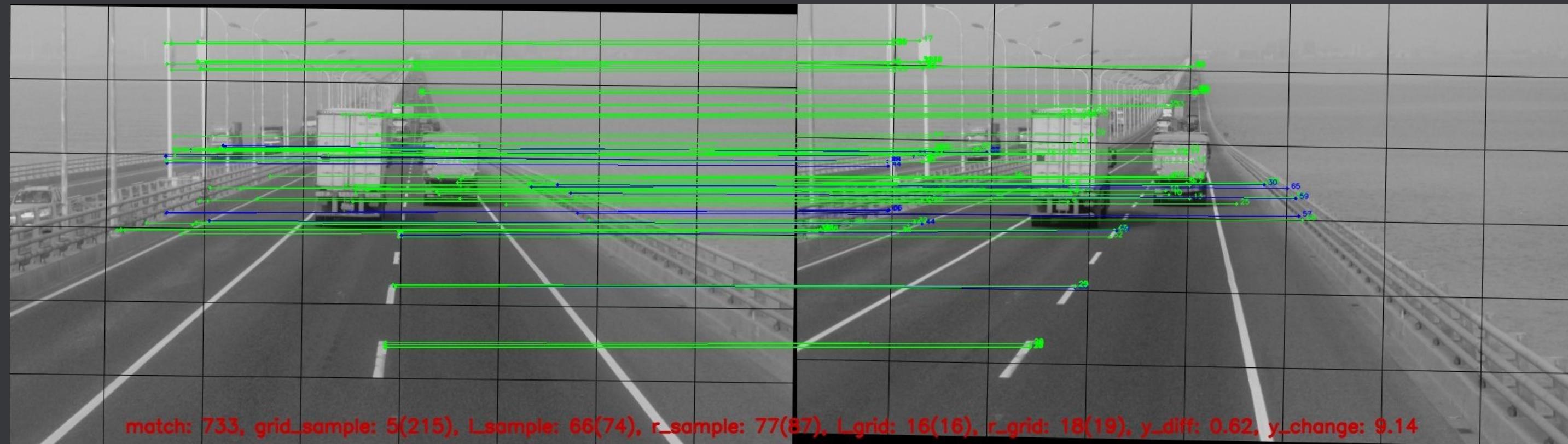
Ceres with selected weighted points

Lidar-based Implementation

2 Pseudo Rectification



OpenCV rectification pattern



Pseudo rectification pattern

Lidar-based Implementation

2 Pseudo Rectification



CFD

Lidar-based Implementation

2 Pseudo Rectification



OpenCV Rectification

Lidar-based Implementation

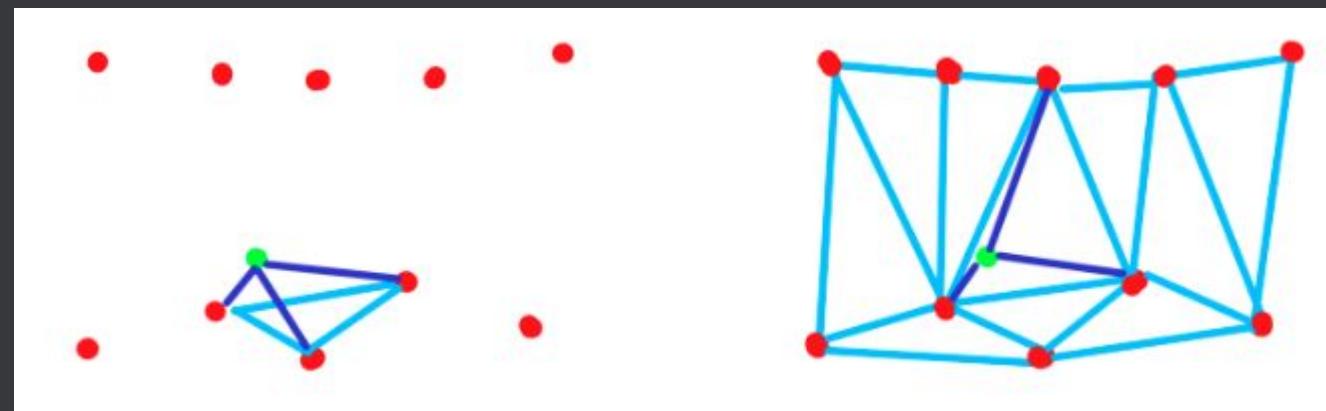
2 Pseudo Rectification



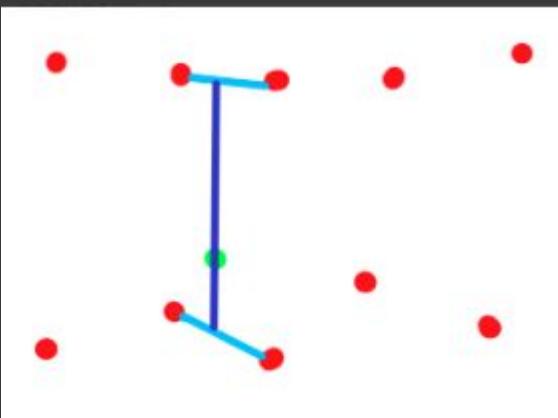
Pseudo Rectification

Lidar-based Implementation

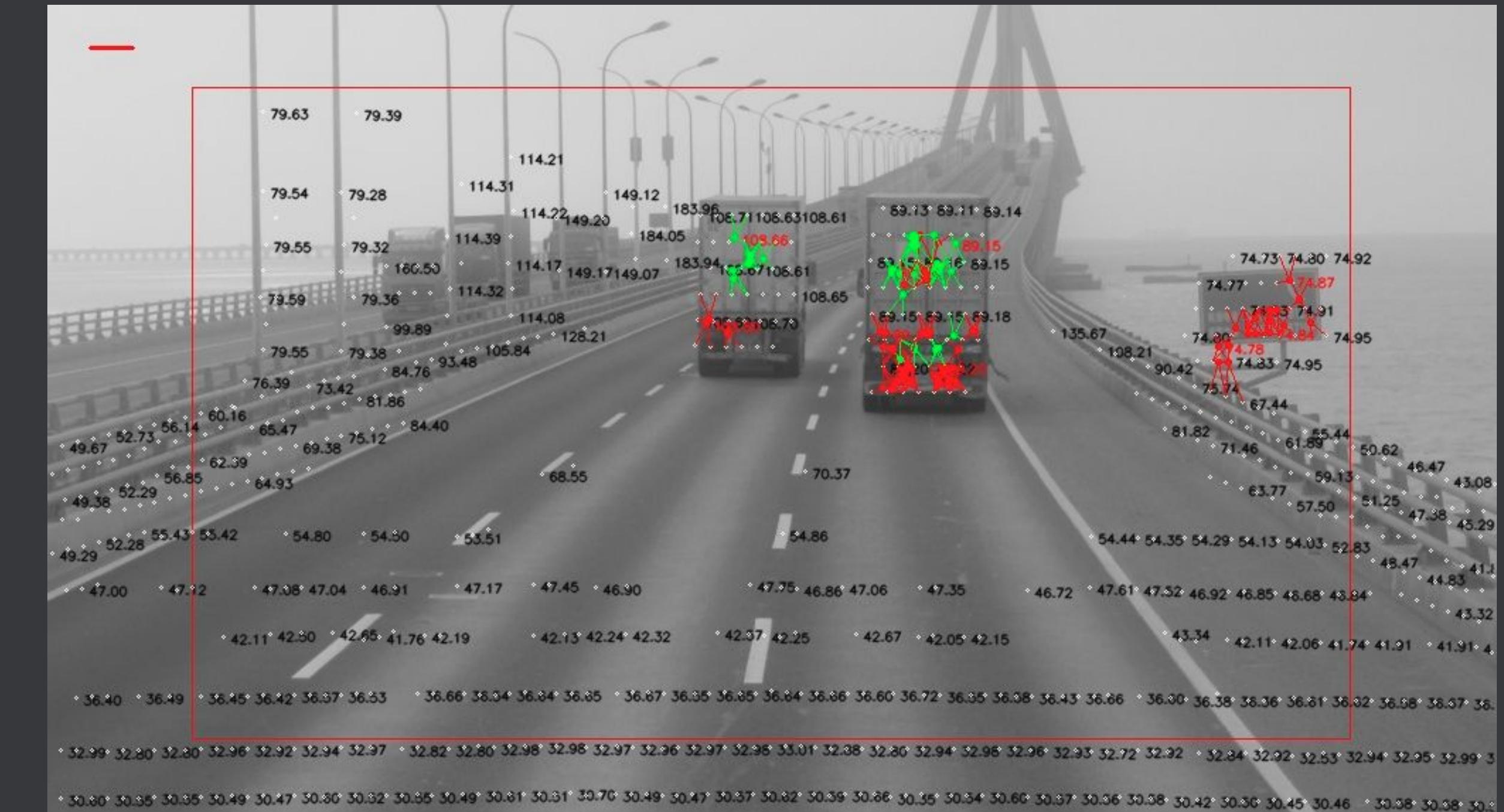
3 Lidar Association



KD-tree and Delaunay triangulation



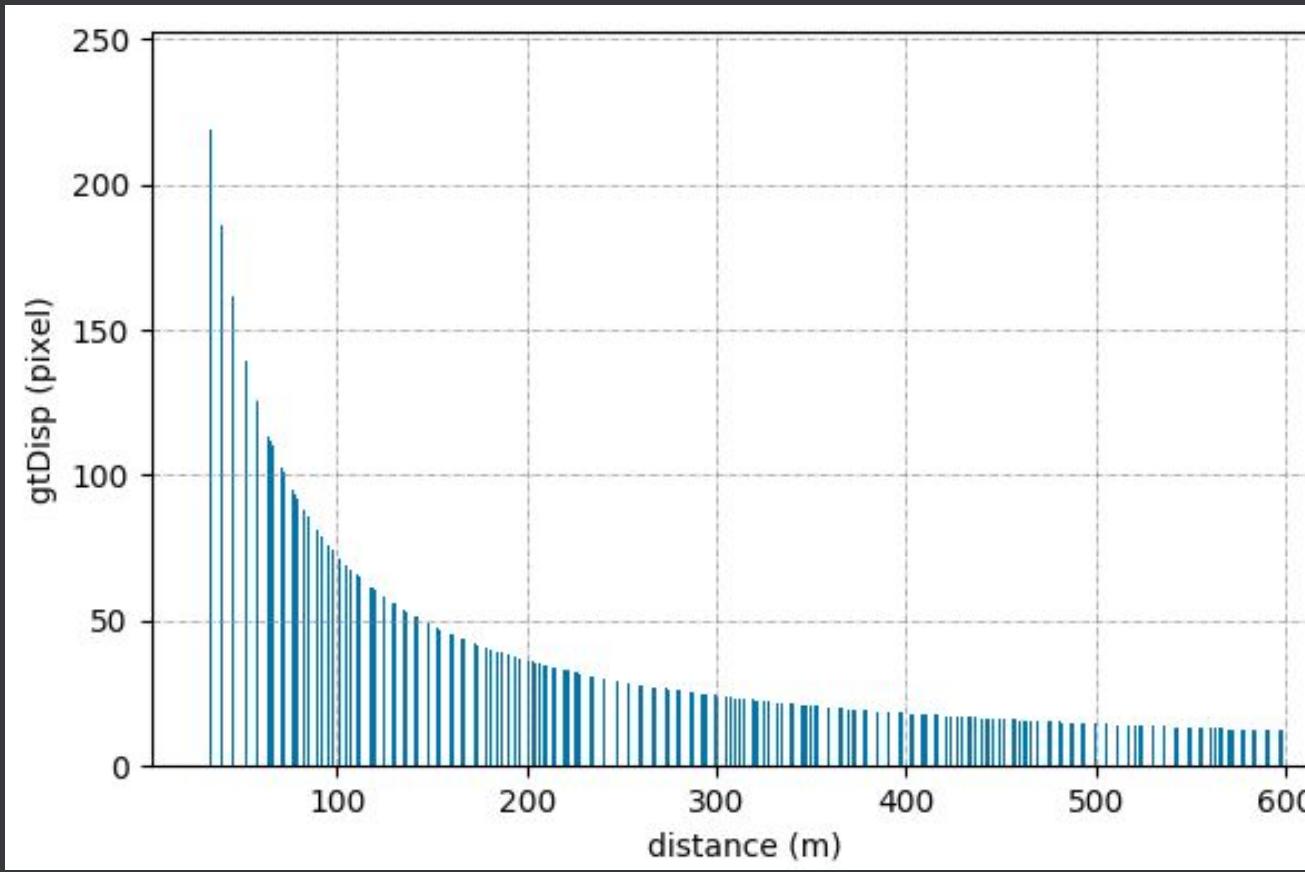
Interpolation from adjacent lidar scans



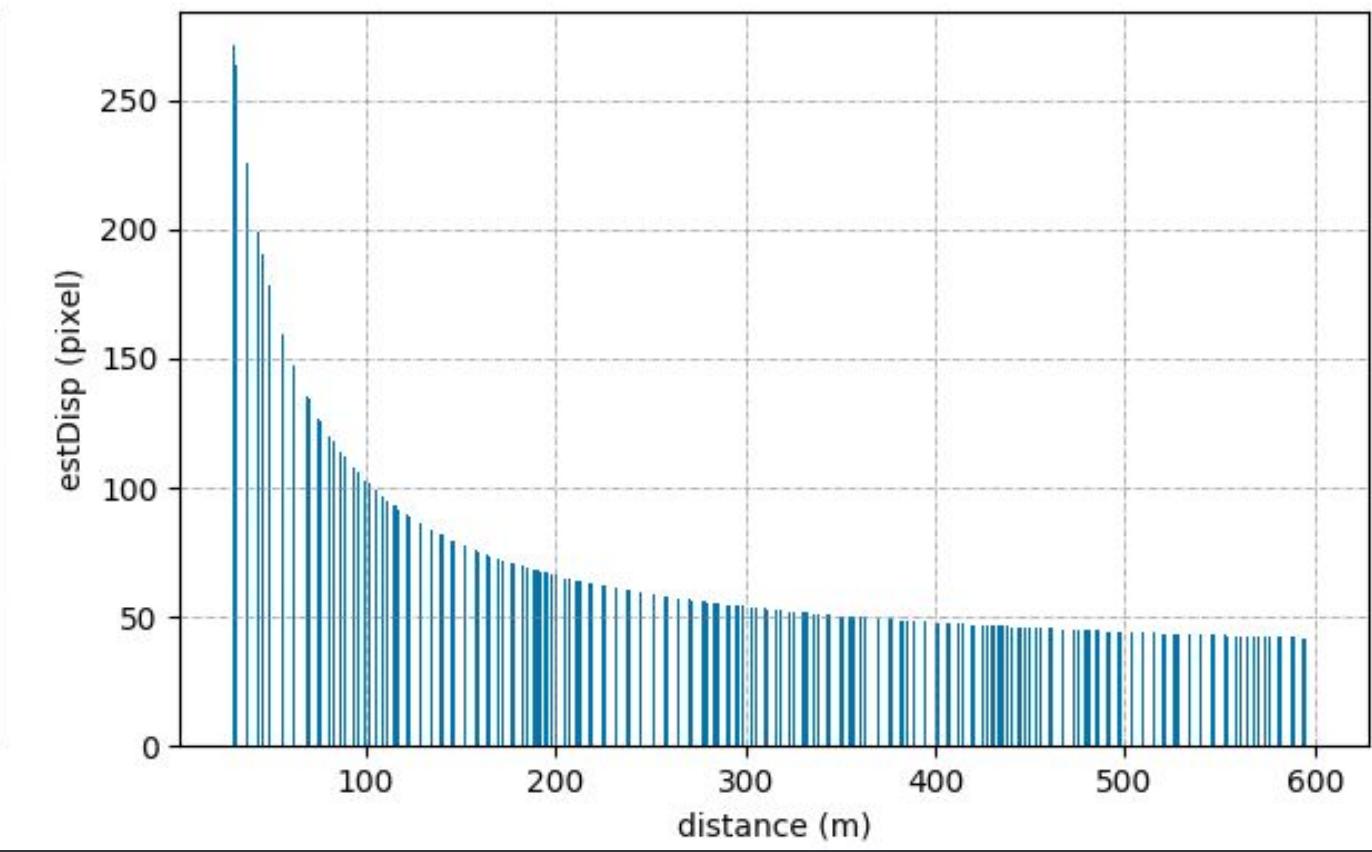
lidar association

Lidar-based Implementation

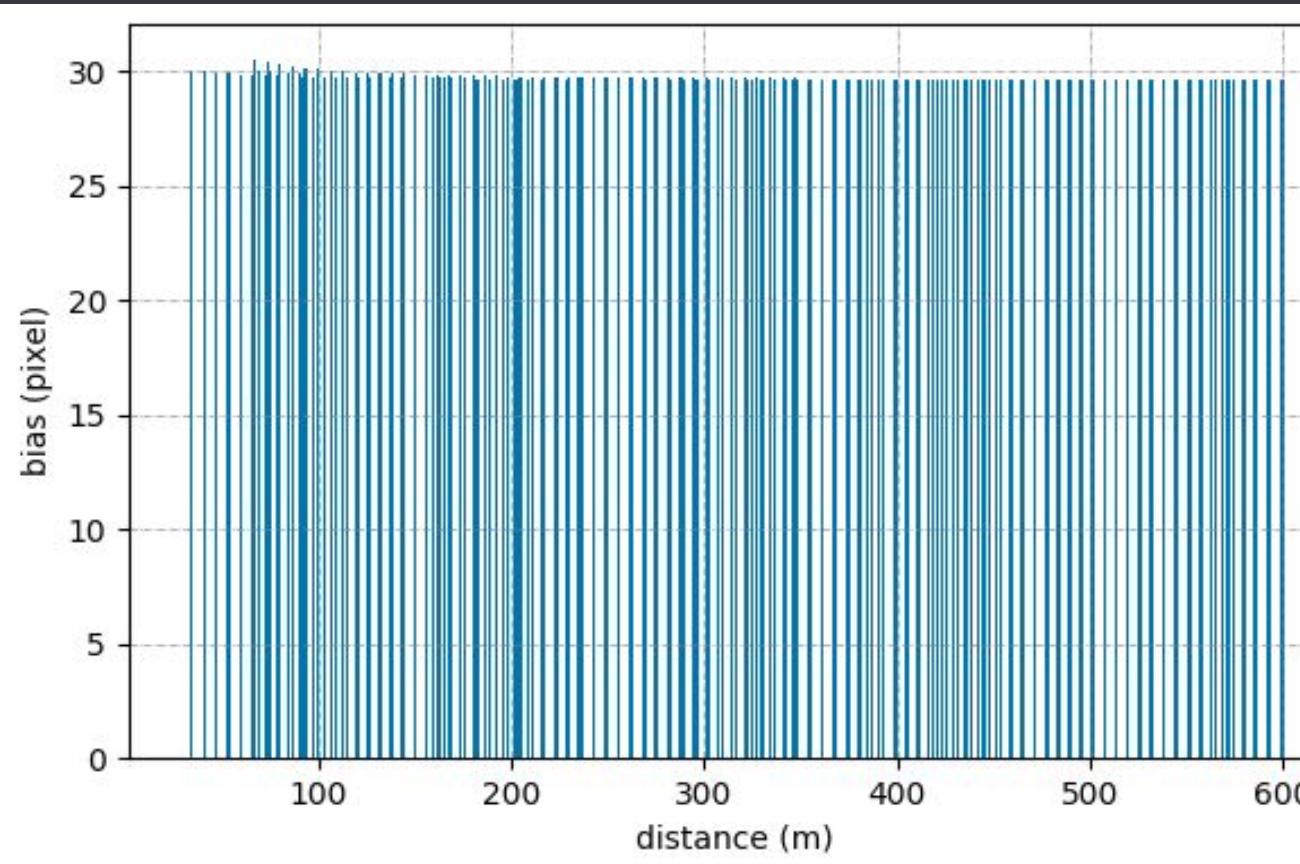
4 Ambiguity Removal



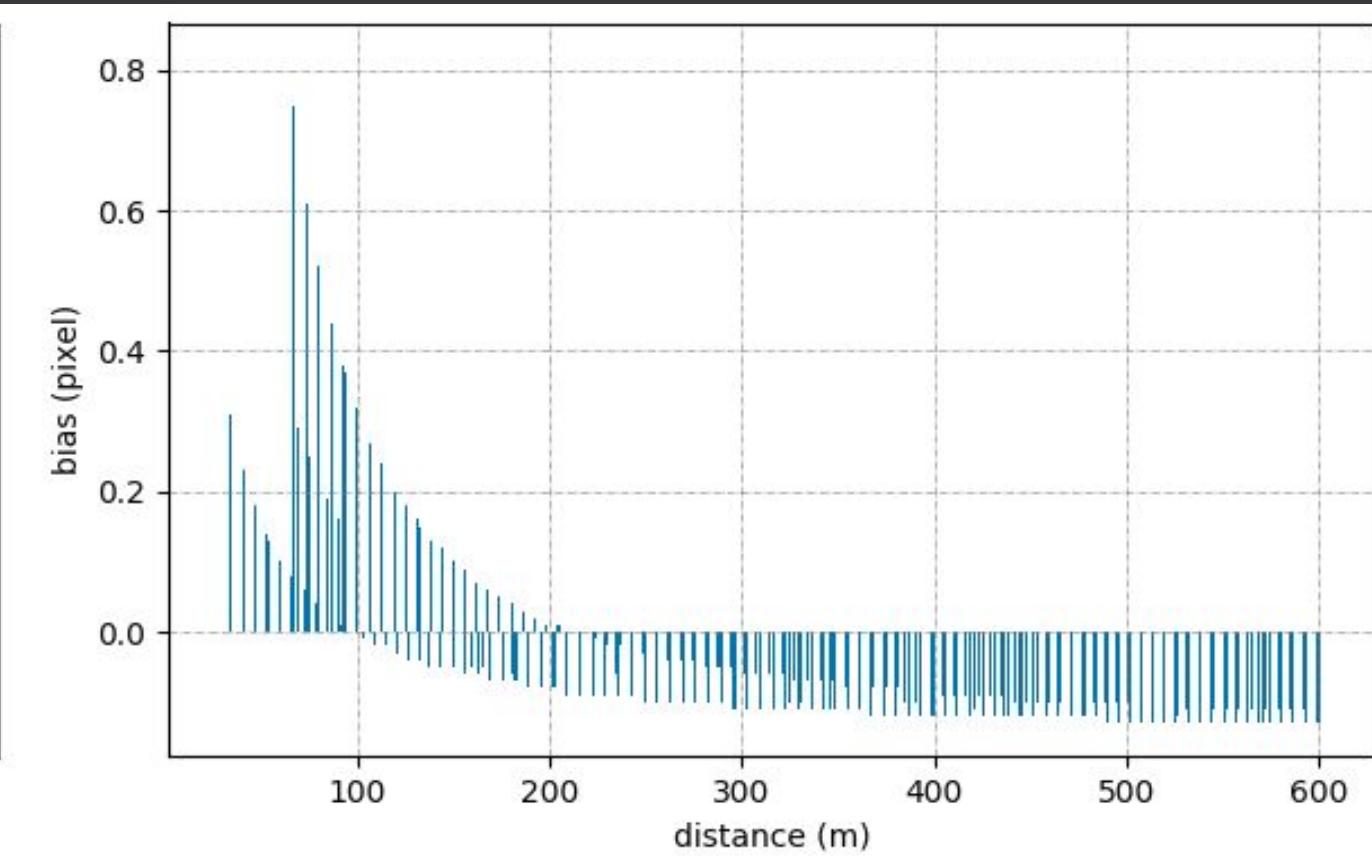
true disparity



estimated disparity

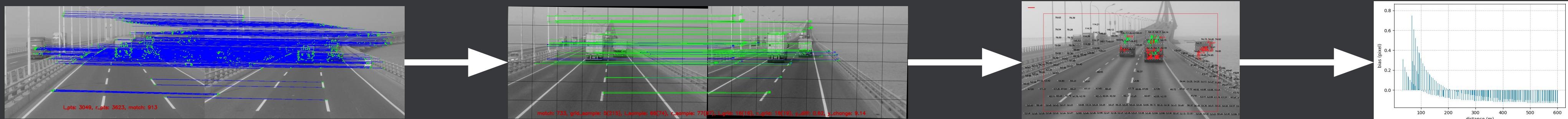


disparity bias



disparity errors after bias removal

Lidar-based Implementation Pipeline



1 feature

2 pseudo rectification

3 lidar association

4 bias removal

```

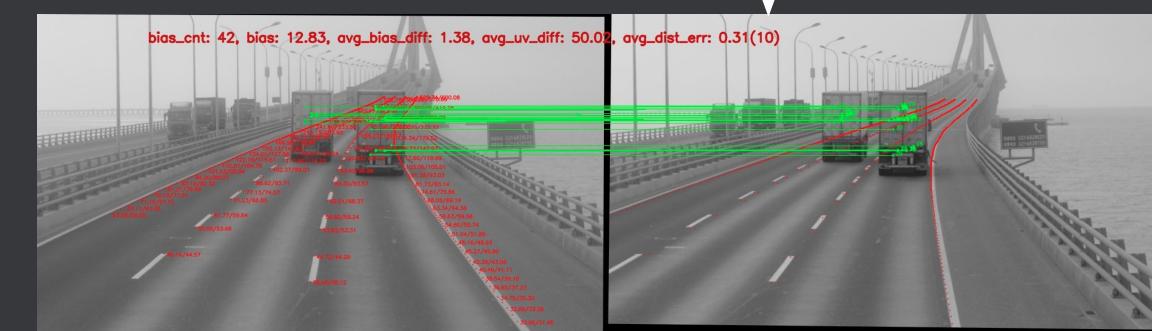
lidar_depth = getDepth(l_u, l_v, l_cam_k, lidar_dist)
gt_disp = l_f * baseline
(rect_u, rect_v) = rectify(u, v, Affine)
est_disp = l_rect_u - l_cx + r_cx - r_rect_u
disp_bias = est_disp - gt_disp
Affine' = removeBias(disp_bias, Affine)
    
```

4 bias removal

```

(rect_u, rect_v) = rectify(u, v, Affine)
disp = l_rect_u - l_cx + r_cx - r_rect_u
depth = l_fx * baseline
dist = getDist(l_rect_u, l_rect_v, l_cam_k, depth)
    
```

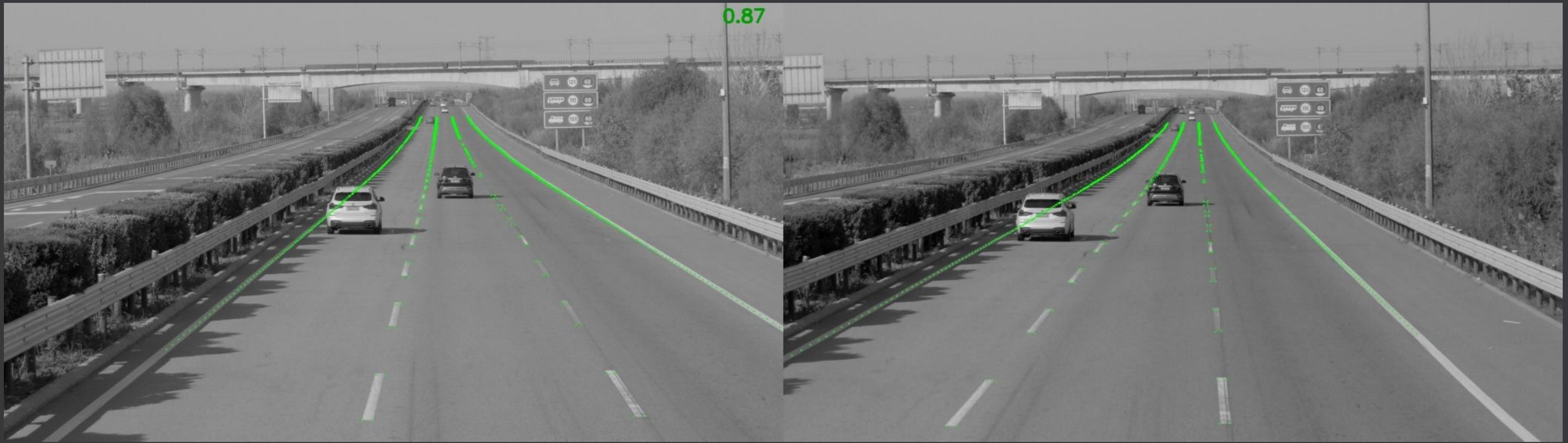
5 depth sensing



5 depth sensing

Lidar-based Implementation

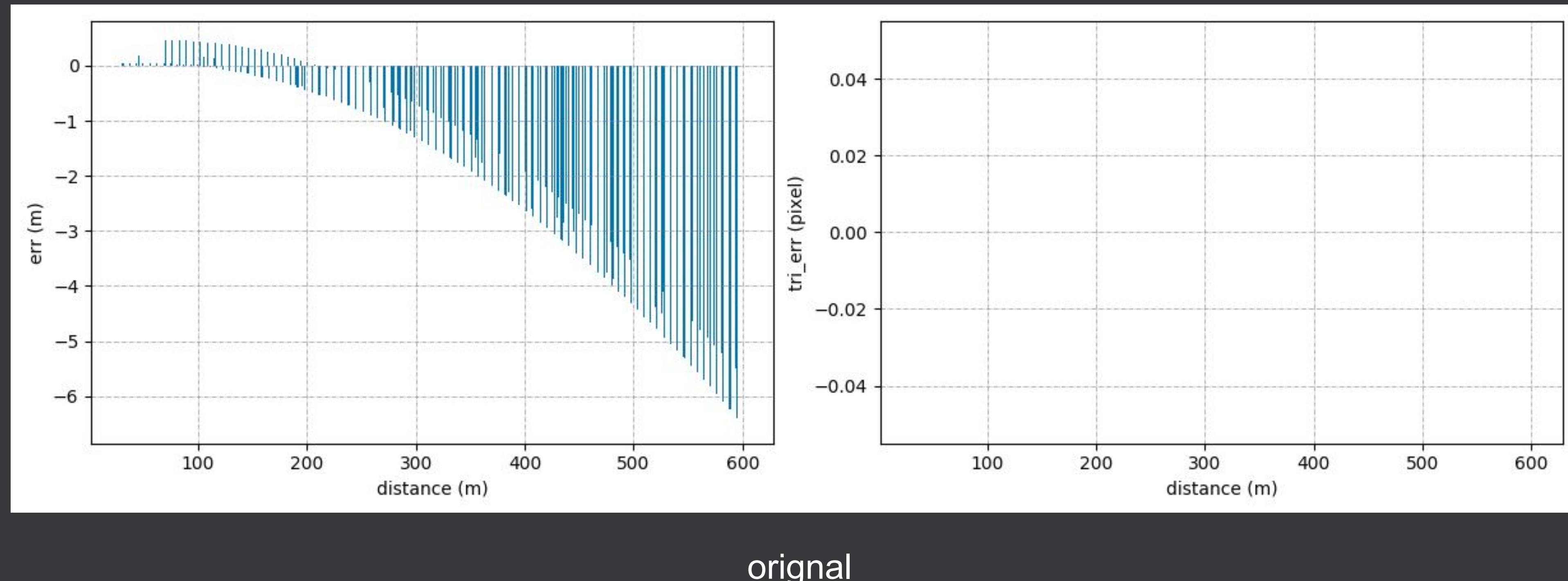
Cam Intrinsics Error



original

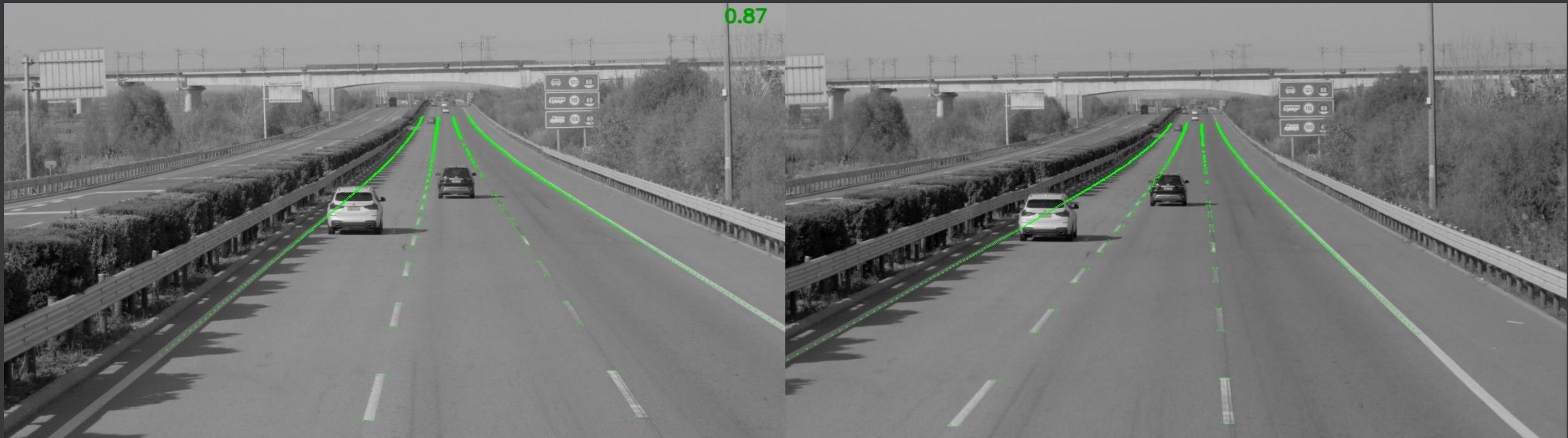
Lidar-based Implementation

Cam Intrinsics Error



Lidar-based Implementation

Cam Intrinsics Error



f_x/f_y in right cam with -30 pixels errors

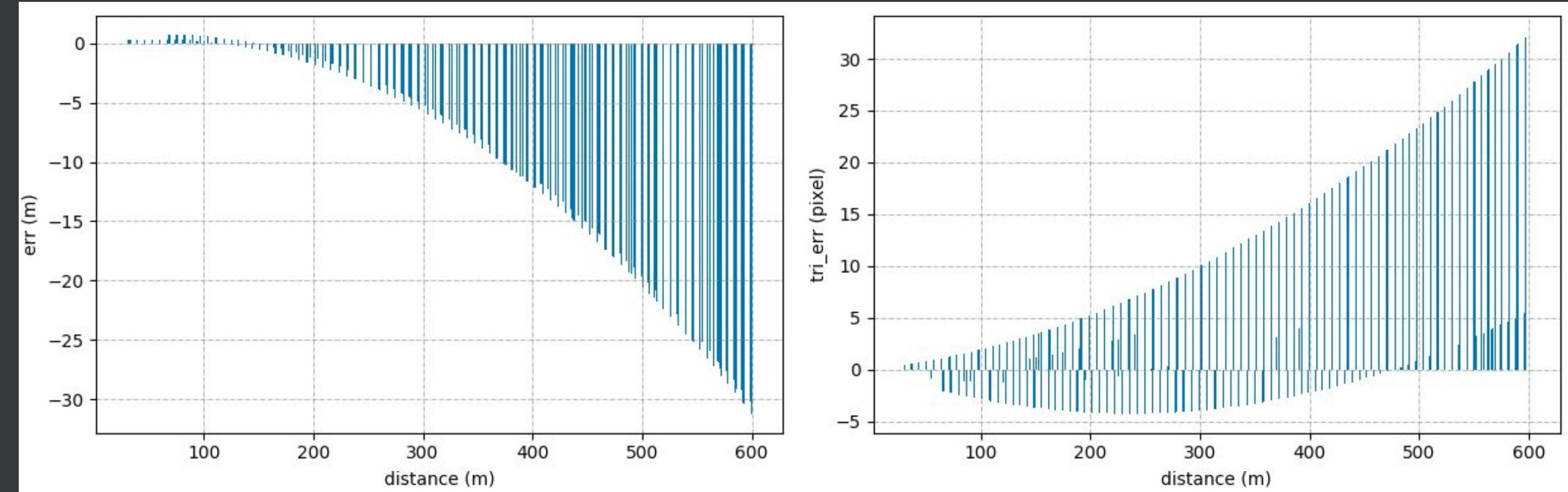
Lidar-based Implementation

Cam Intrinsics Error

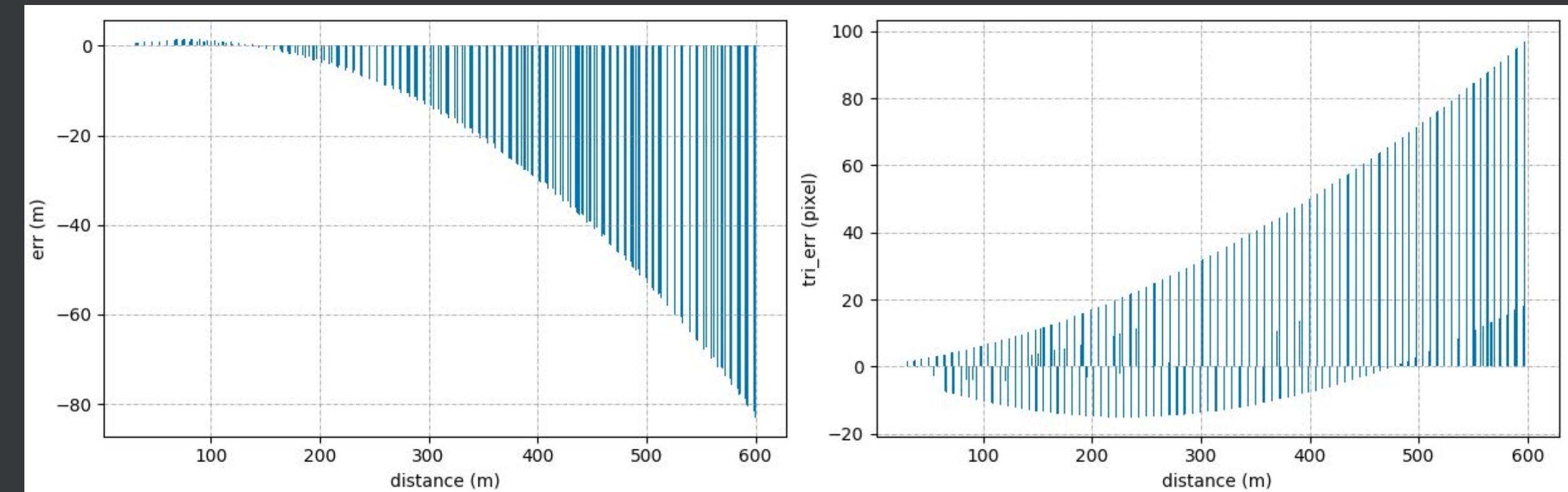


f_x/f_y in right cam with -300 pixels errors

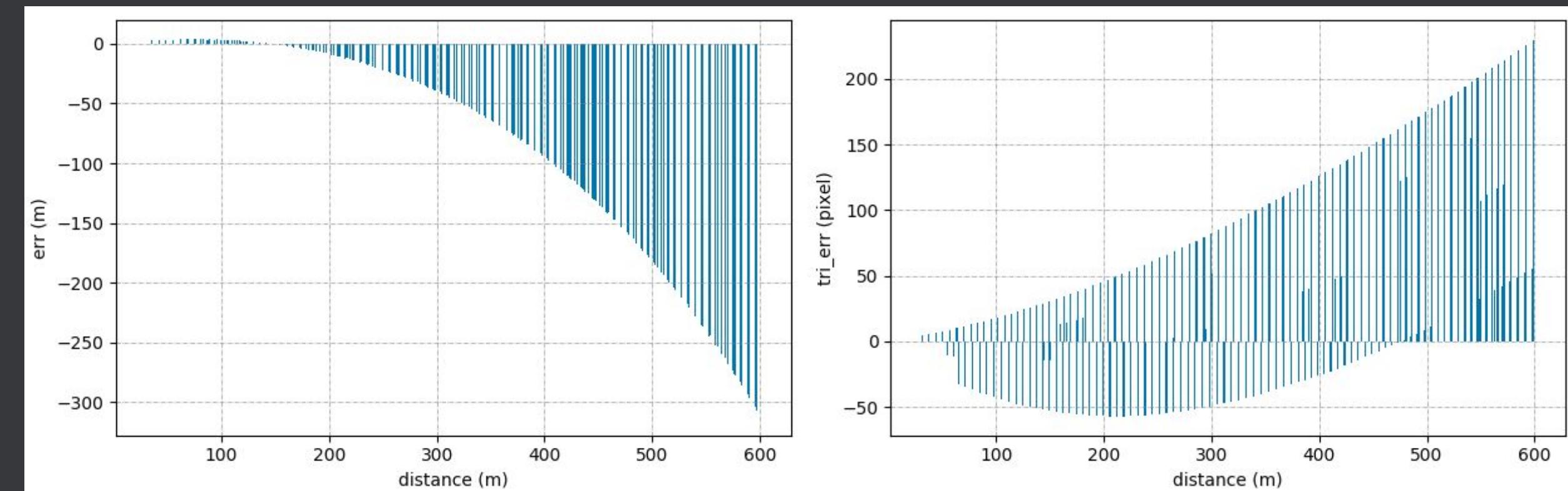
fx/fy in left cam with -30 pixels errors



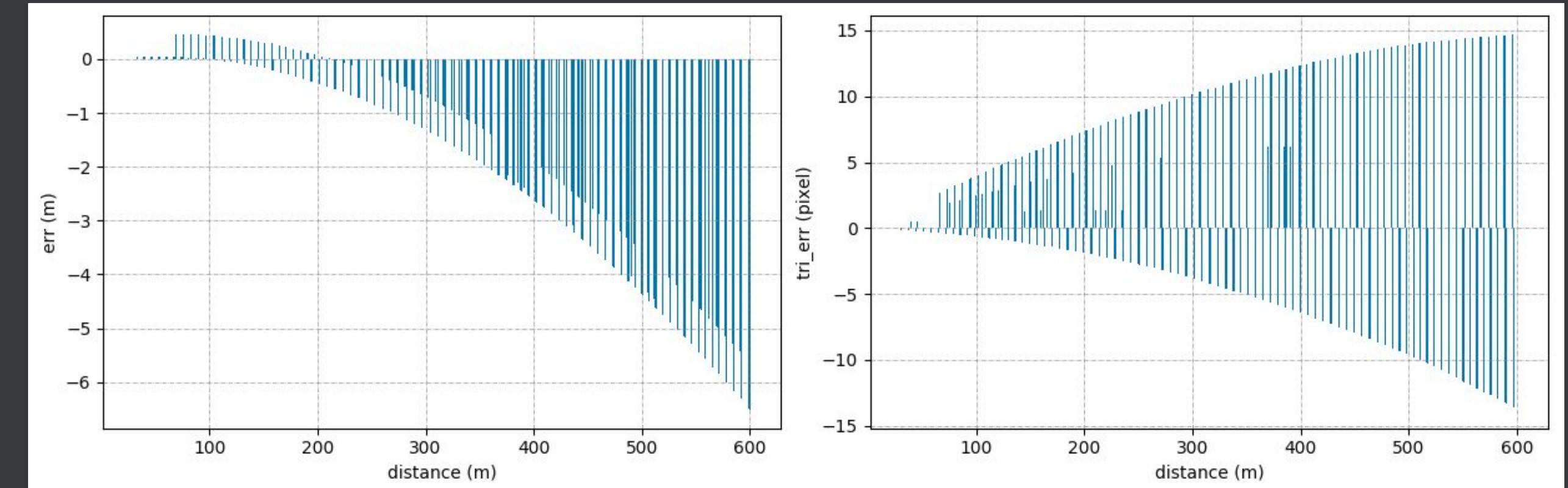
fx/fy in left cam with -100 pixels errors



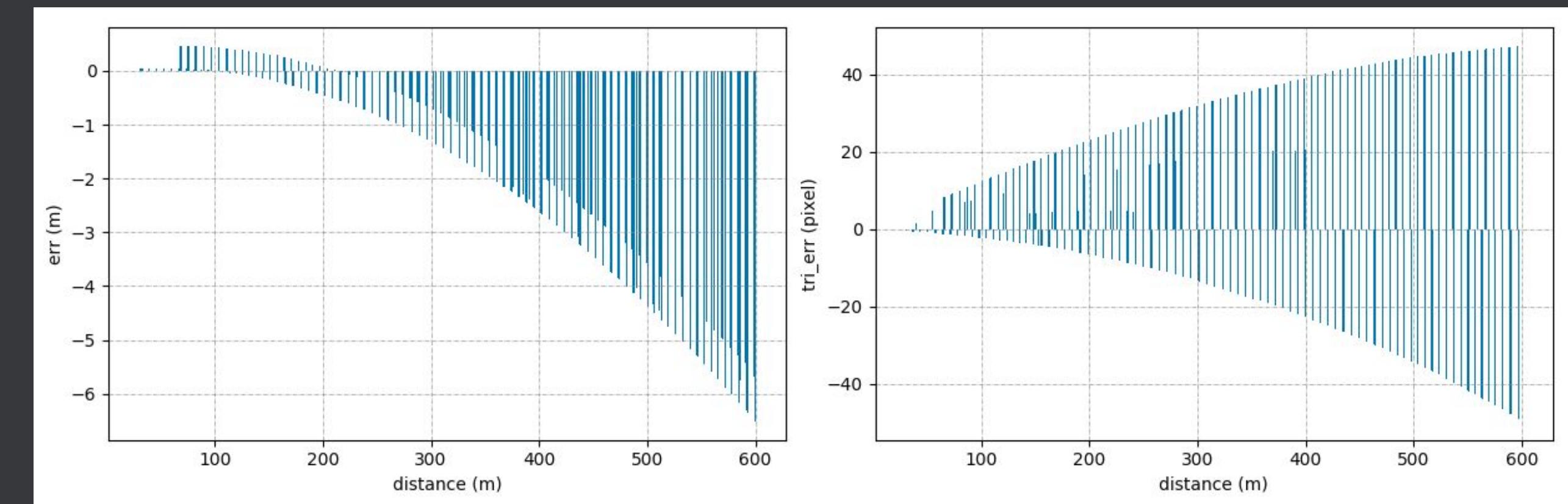
fx/fy in left cam with -300 pixels errors



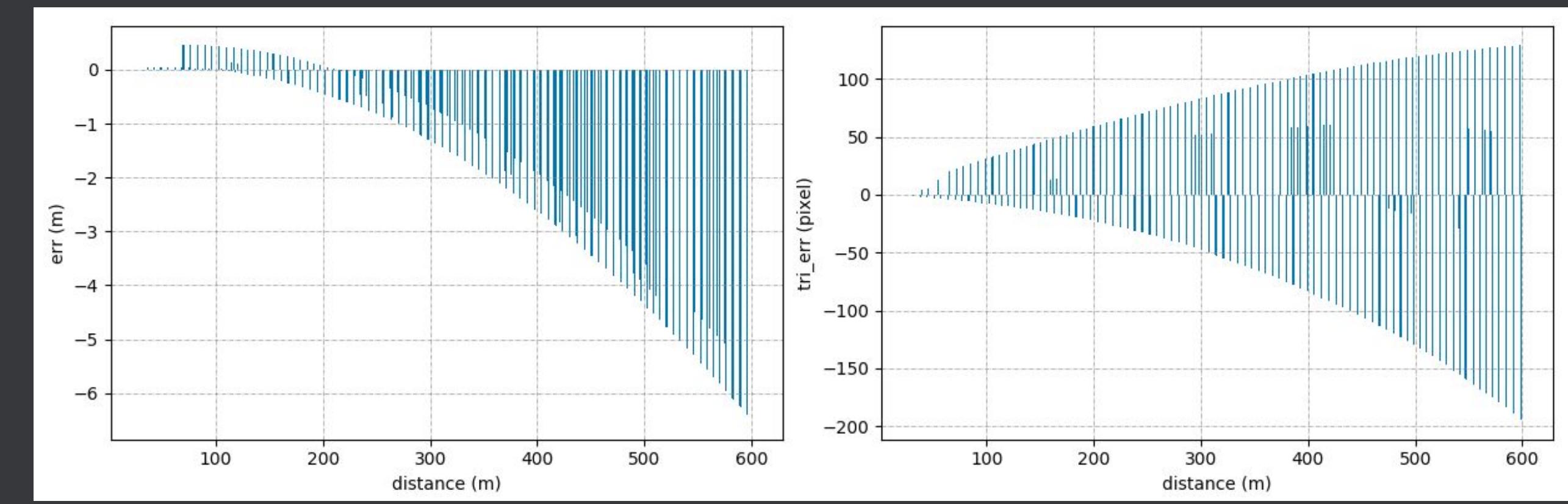
fx/fy in right cam with -30 pixels errors



fx/fy in right cam with -100 pixels errors

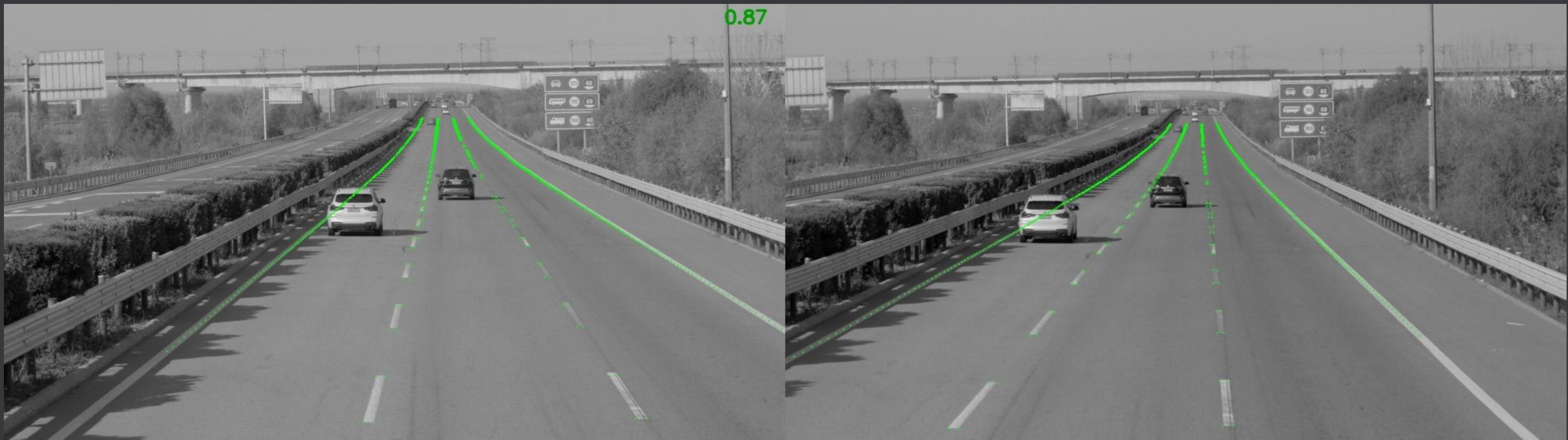


fx/fy in right cam with -300 pixels errors



Lidar-based Implementation

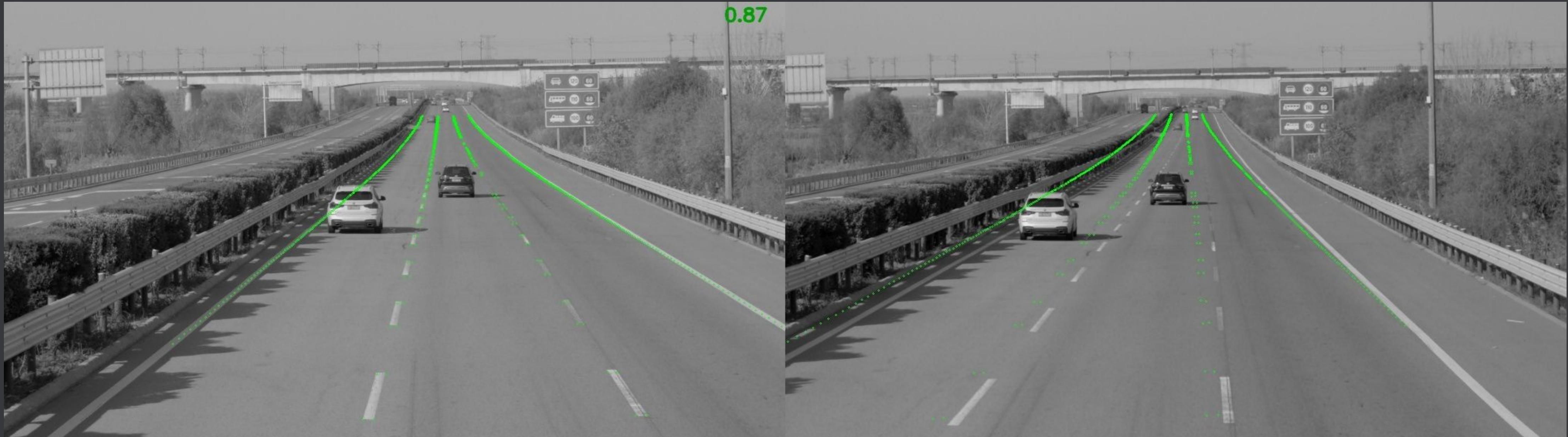
Cam Intrinsics Error



cx/cy in right cam with -0.2/0.1 pixels errors

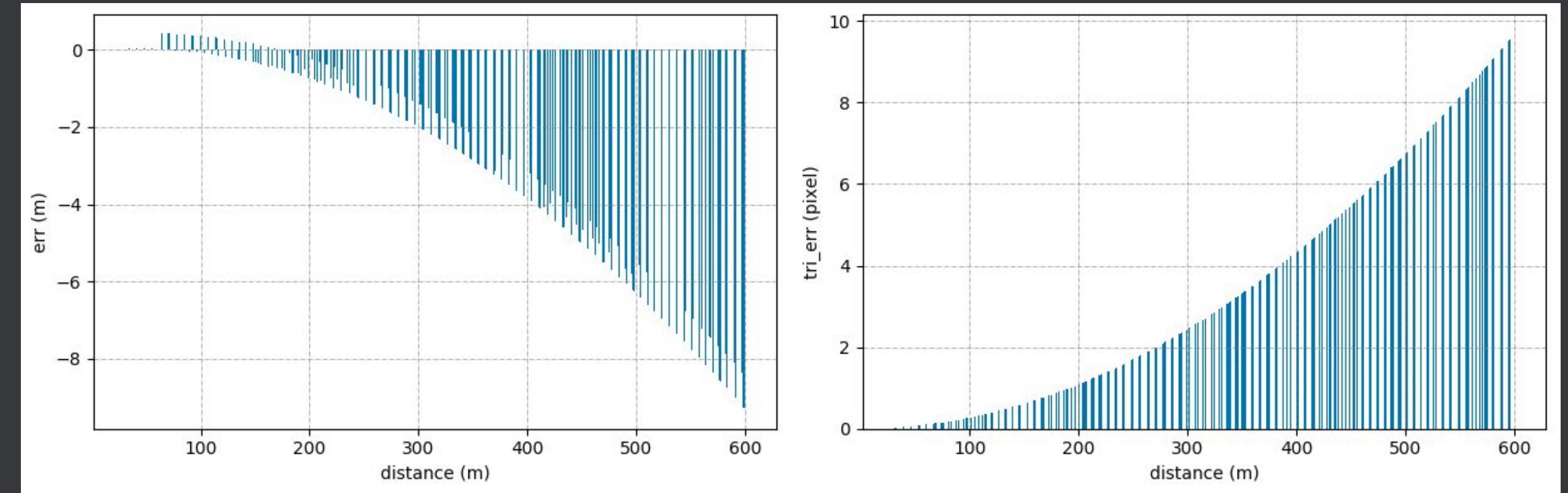
Lidar-based Implementation

Cam Intrinsics Error

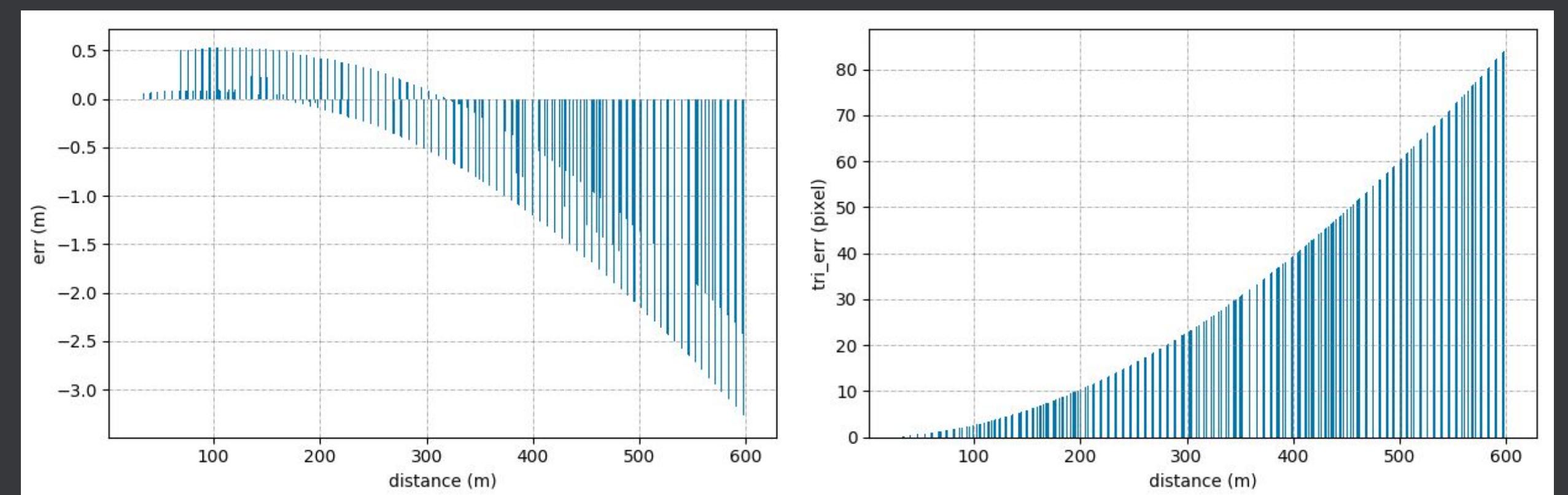


cx/cy in right cam with 20/10 pixels errors

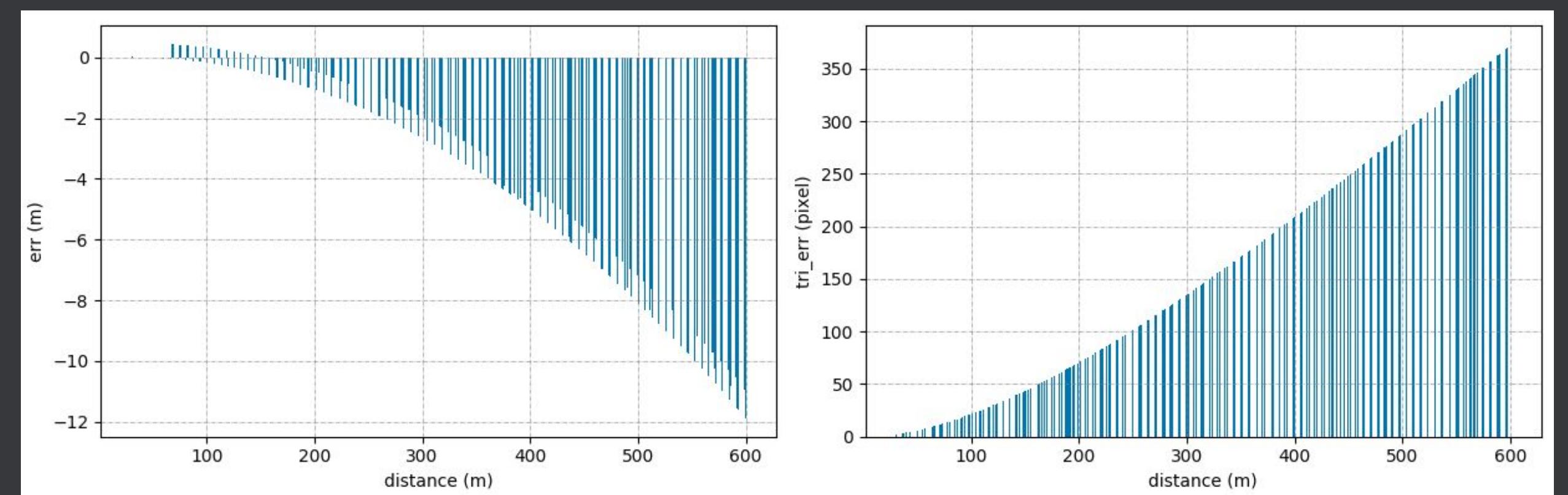
cx/cy in left cam with 0.2/0.1 pixels errors



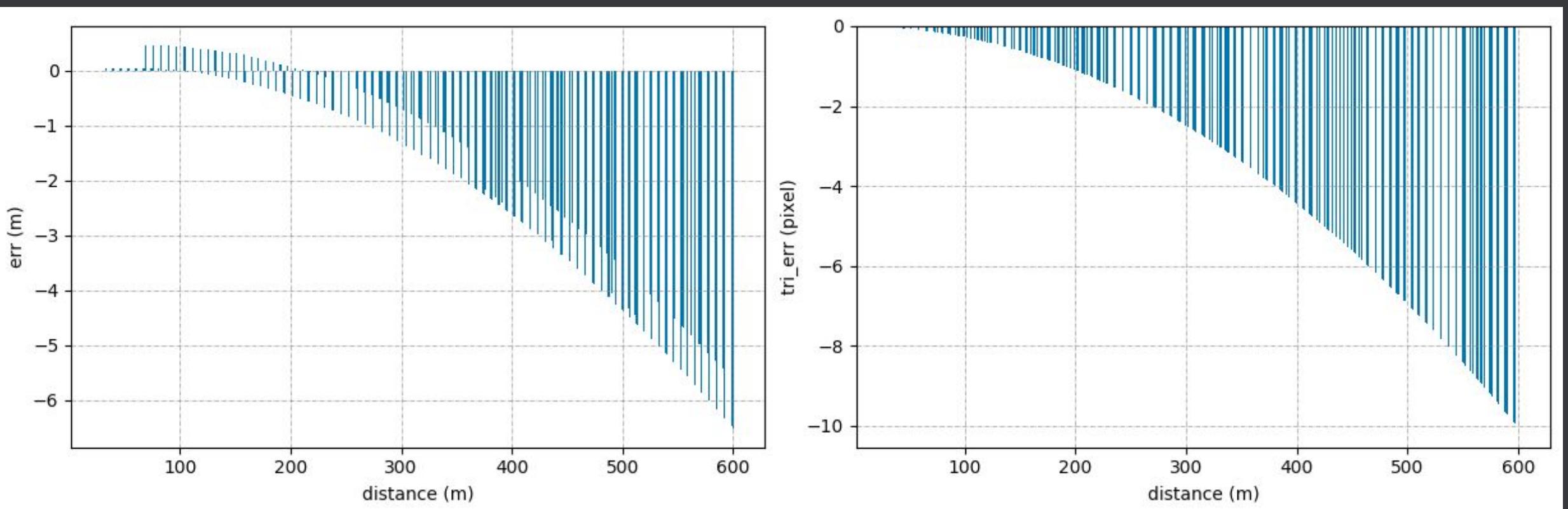
cx/cy in left cam with 2/1 pixels errors



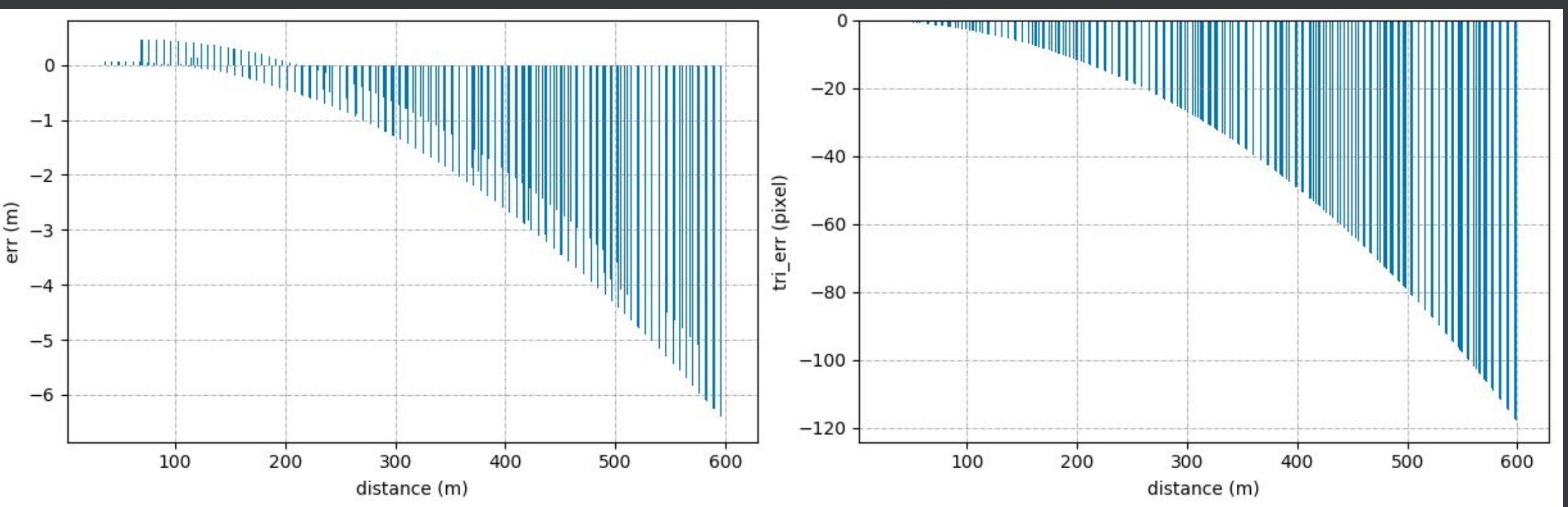
cx/cy in left cam with 20/10 pixels errors



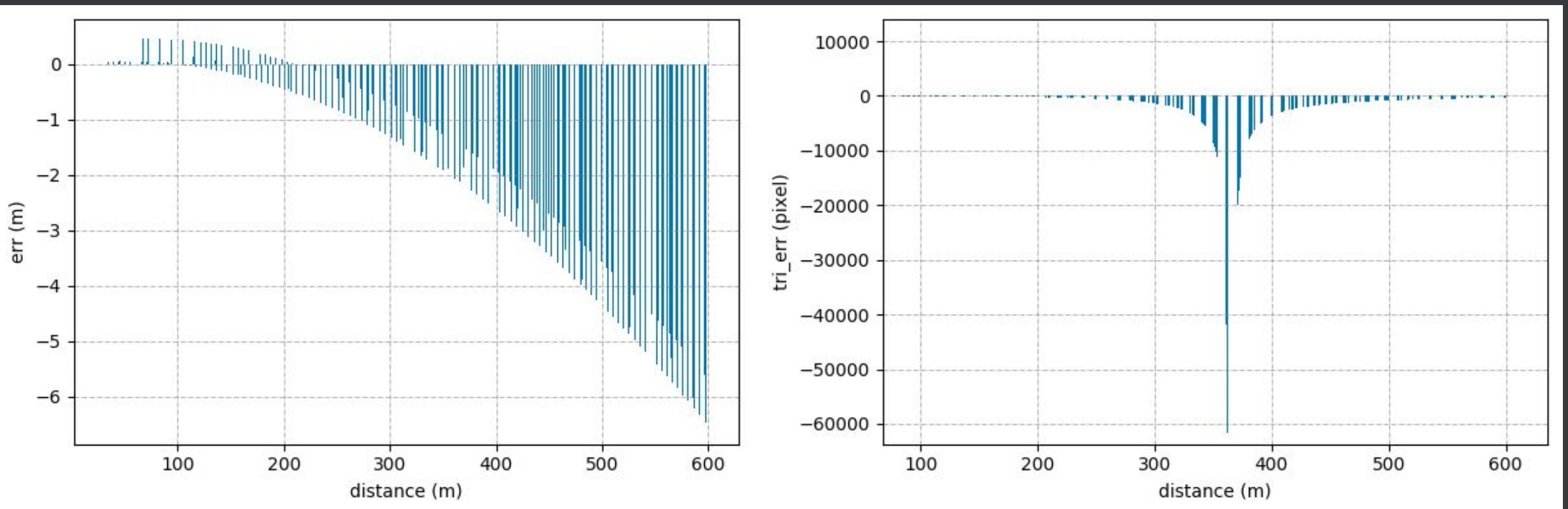
cx/cy in right cam with 0.2/0.1 pixels errors



cx/cy in right cam with 2/1 pixels errors

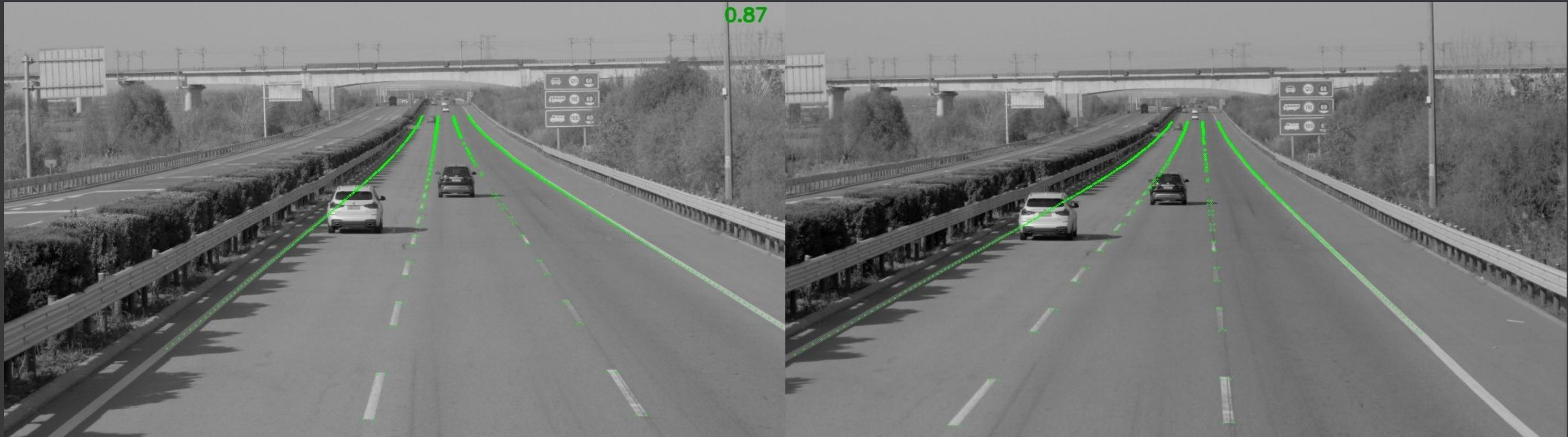


cx/cy in right cam with 20/10 pixels errors



Lidar-based Implementation

Cam extrinsics Error



yaw with -0.02° error

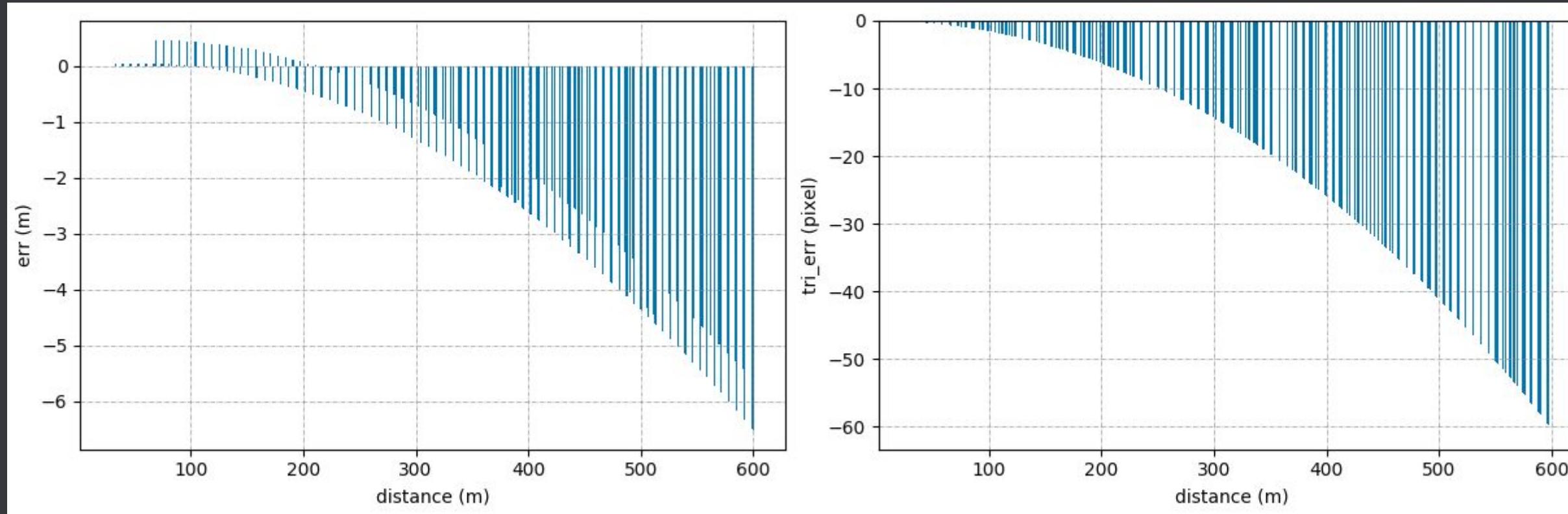
Lidar-based Implementation

Cam extrinsics Error

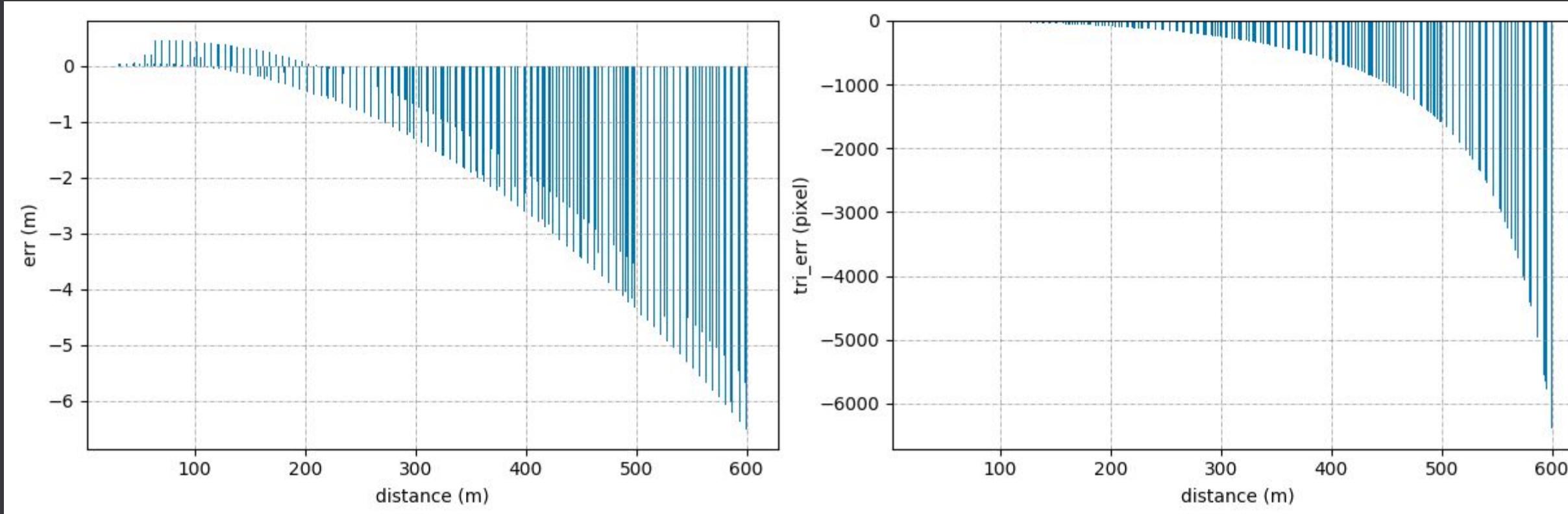


yaw with -0.2° error

yaw with -0.02° error

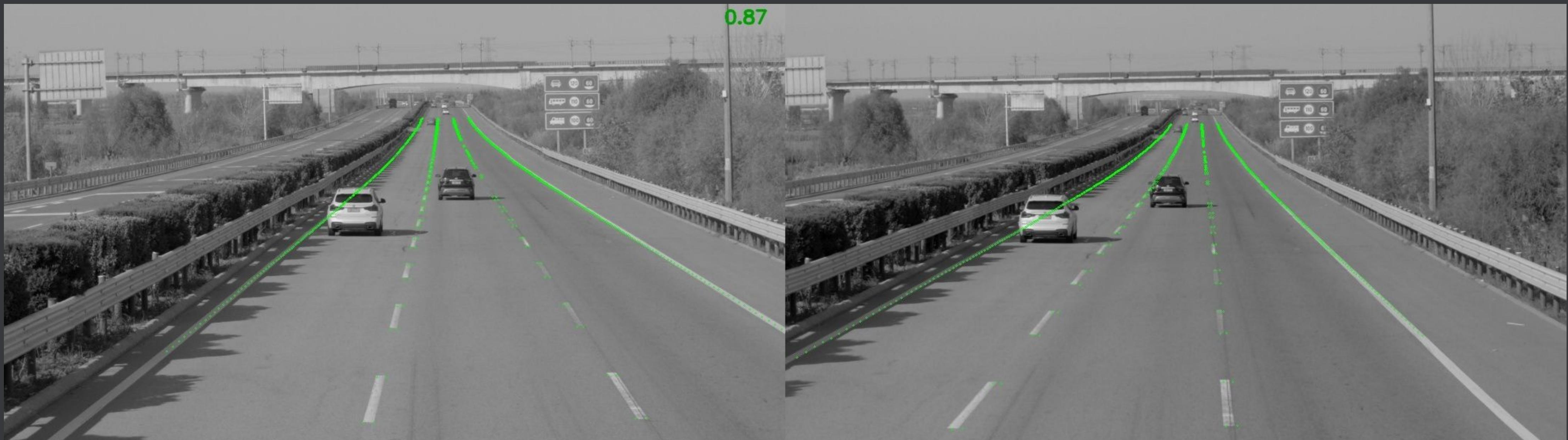


yaw with -0.2° error



Lidar-based Implementation

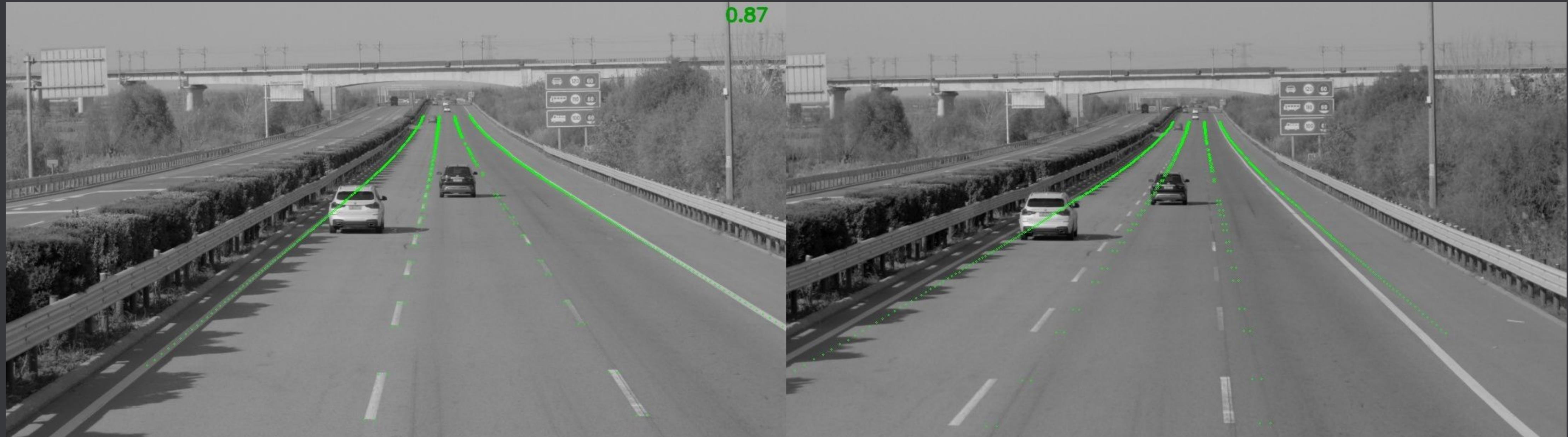
Cam extrinsics Error



baseline with -0.05m error

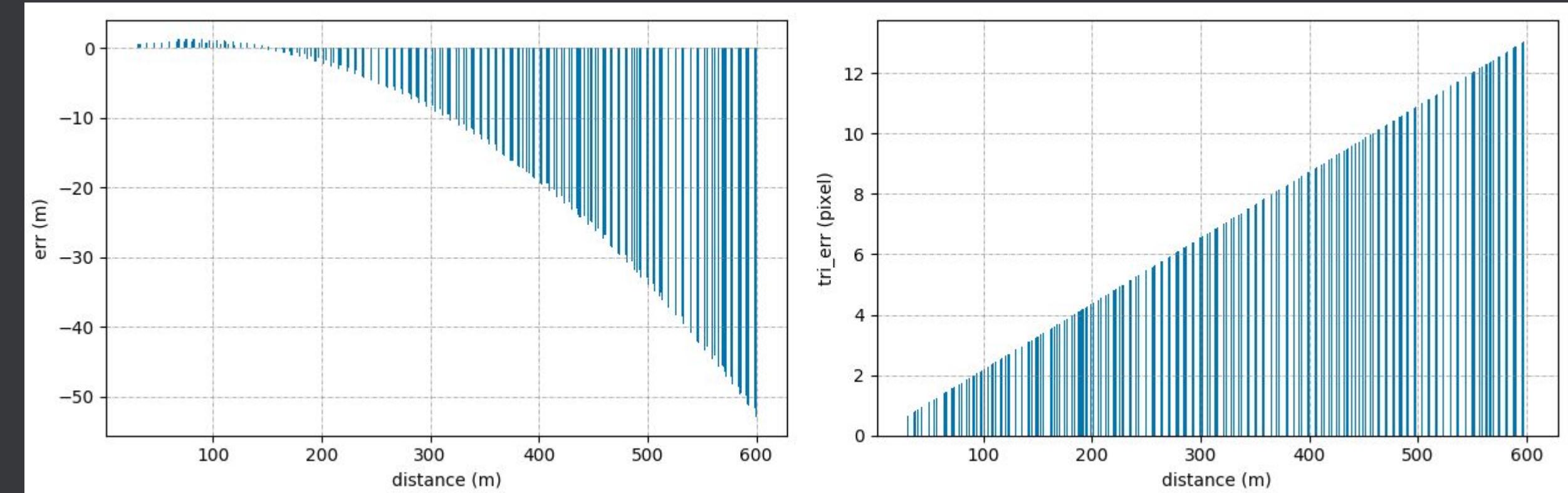
Lidar-based Implementation

Cam extrinsics Error

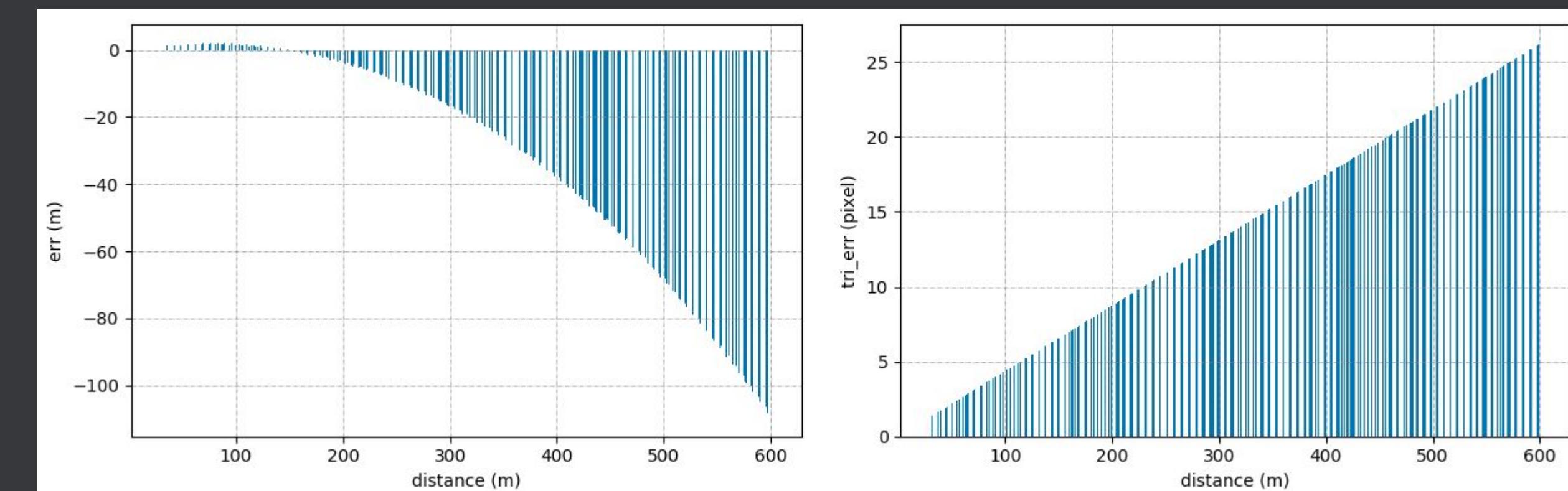


baseline with -0.5m error

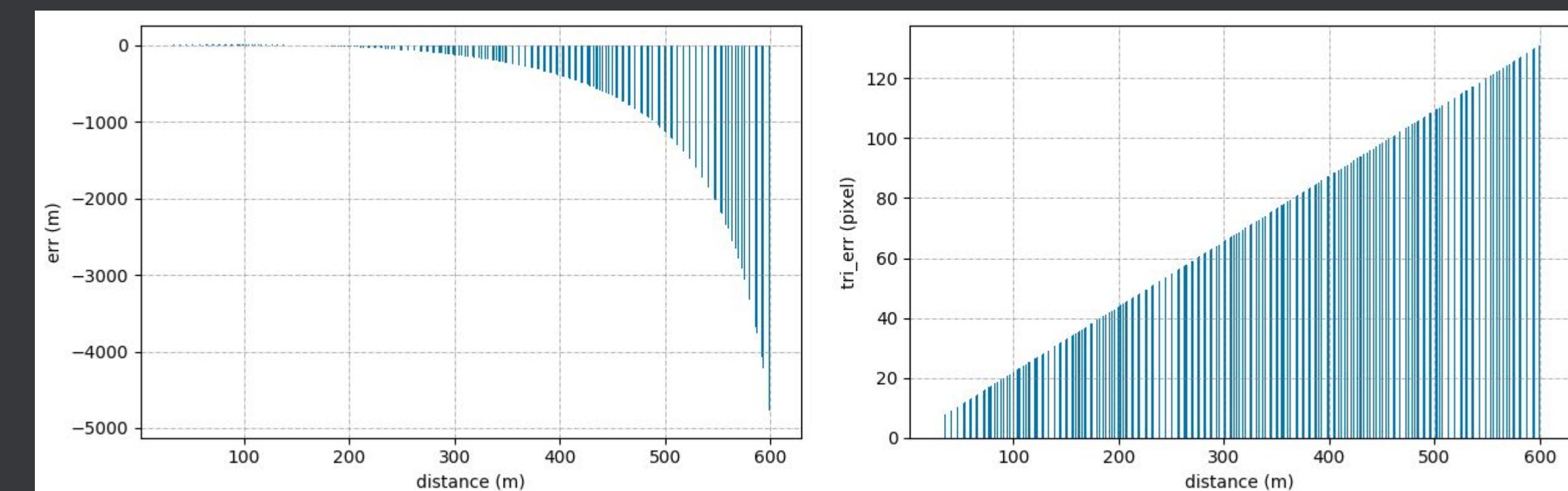
baseline with -0.05m error



baseline with -0.1m error

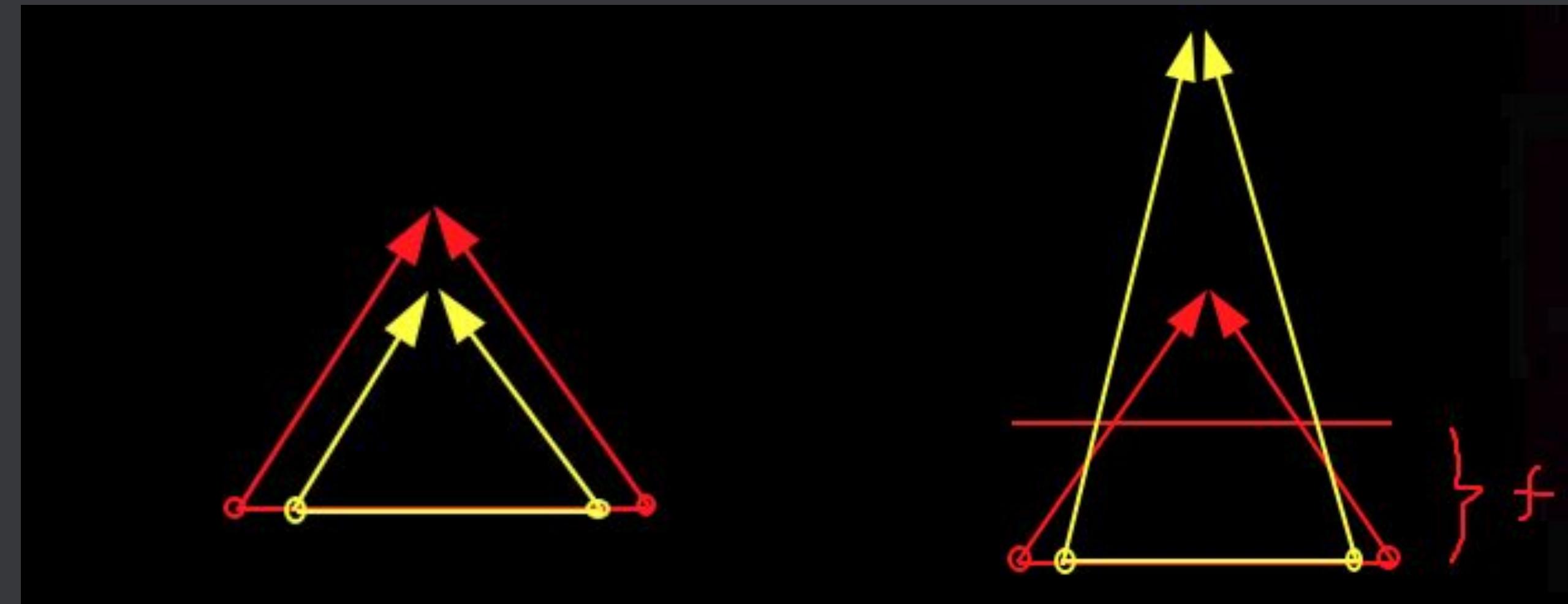


baseline with -0.5m error



Lidar-based Implementation

Cam extrinsics Error



triangulation

paper

Conclusion

Depth Sensing Beyond LiDAR Range in CVPR 2020 and our implementation

Features:

1. Online stereo camera pseudo rectification.
2. Depth sensing requires only partial camera calibration.
3. Need accurate stereo baseline calibration.



Thank You!

