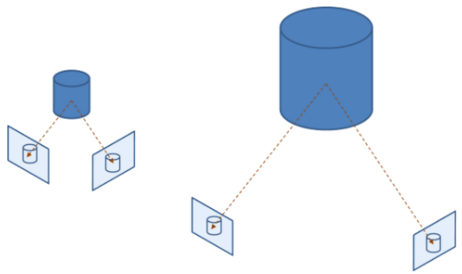
RAFSet 3D-2D motion estimation

part 1. LiDAR interpolation

Jeon Hyun Ho

Intro

- **Frame to frame motion estimation method**
  - 2D-2D motion estimation (Scale problem)
  - √ 3D-2D motion estimation (minimize image reprojection error)
  - 3D-3D motion estimation (minimize feature position error)



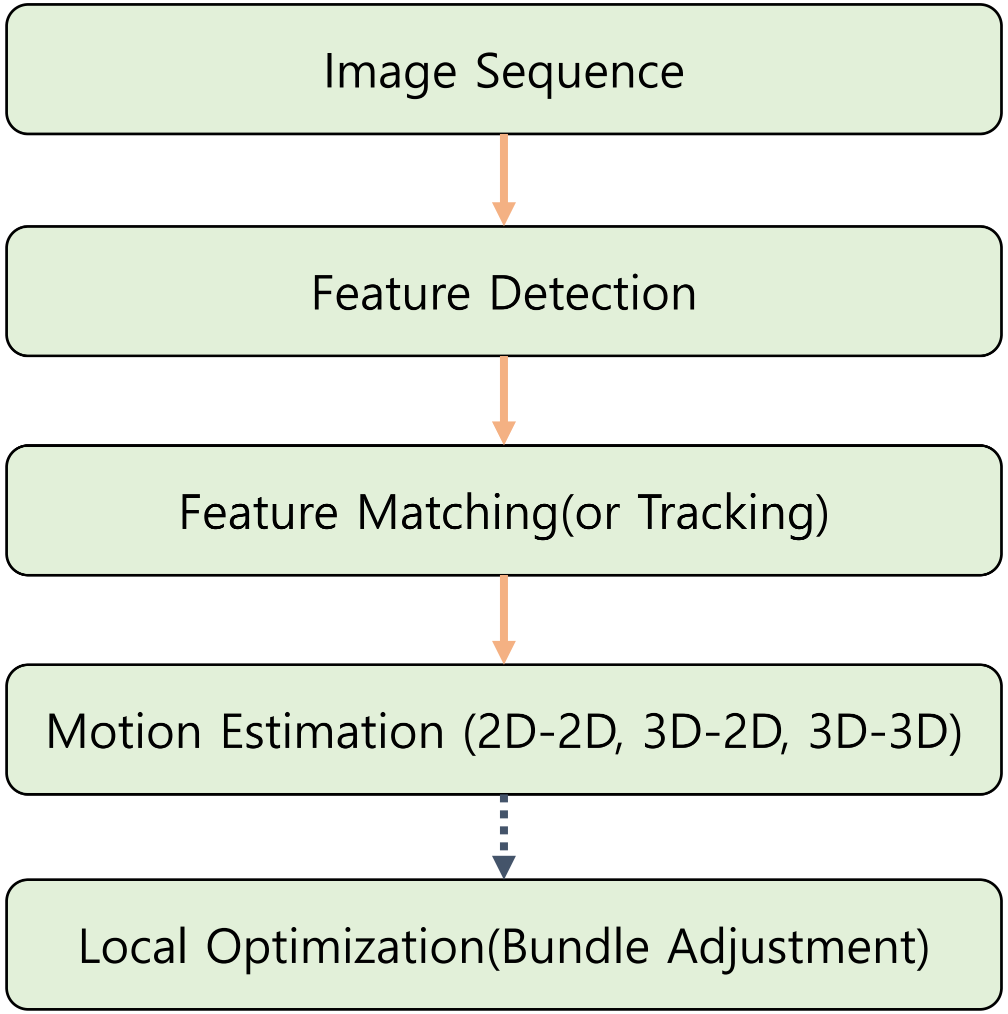
Scale problem

Scaramuzza, Davide, and Friedrich Fraundorfer. "Visual odometry [tutorial]." *IEEE Robotics & Automation Magazine* 18.4 (2011): 80-92.

Fraundorfer, Friedrich, and Davide Scaramuzza. "Visual odometry: Part II: Matching, robustness, optimization, and applications." *IEEE Robotics & Automation Magazine* 19.2 (2012): 78-90.

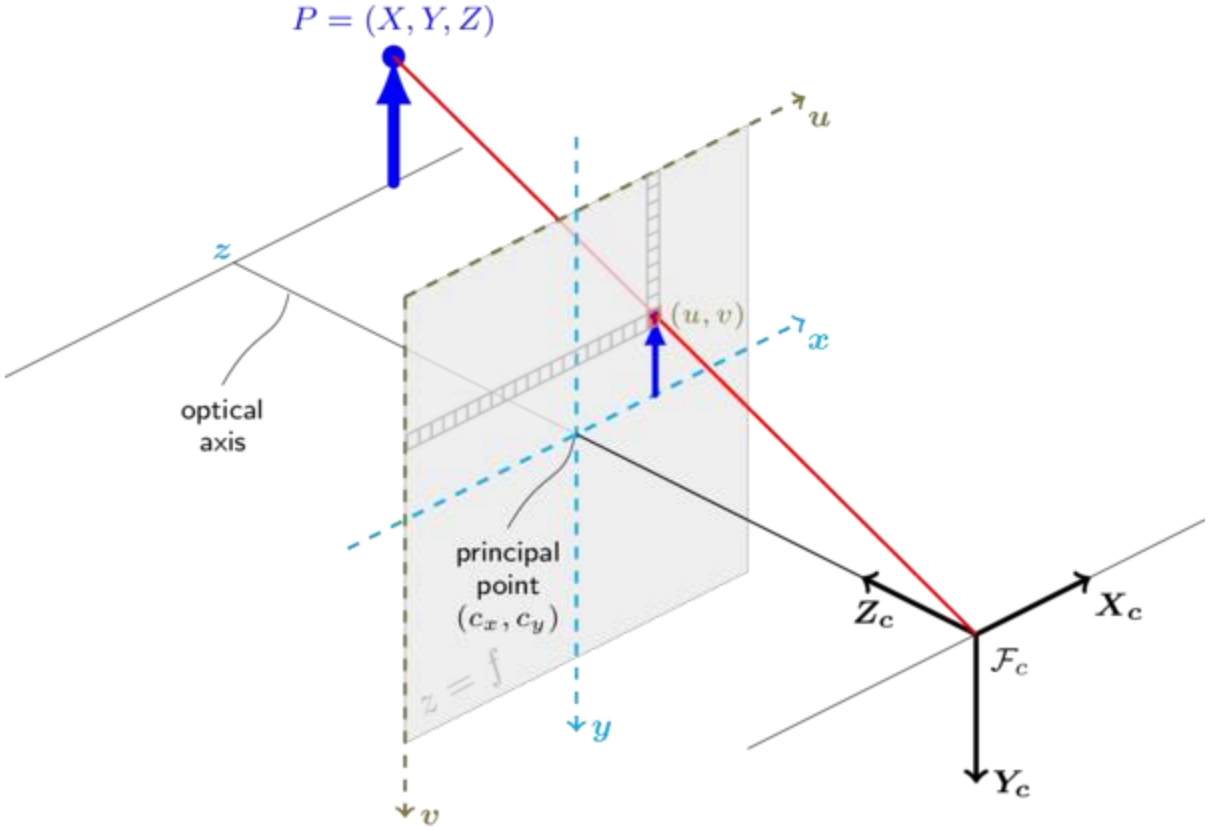
Intro

- **Feature based motion estimation process**



Interpolation

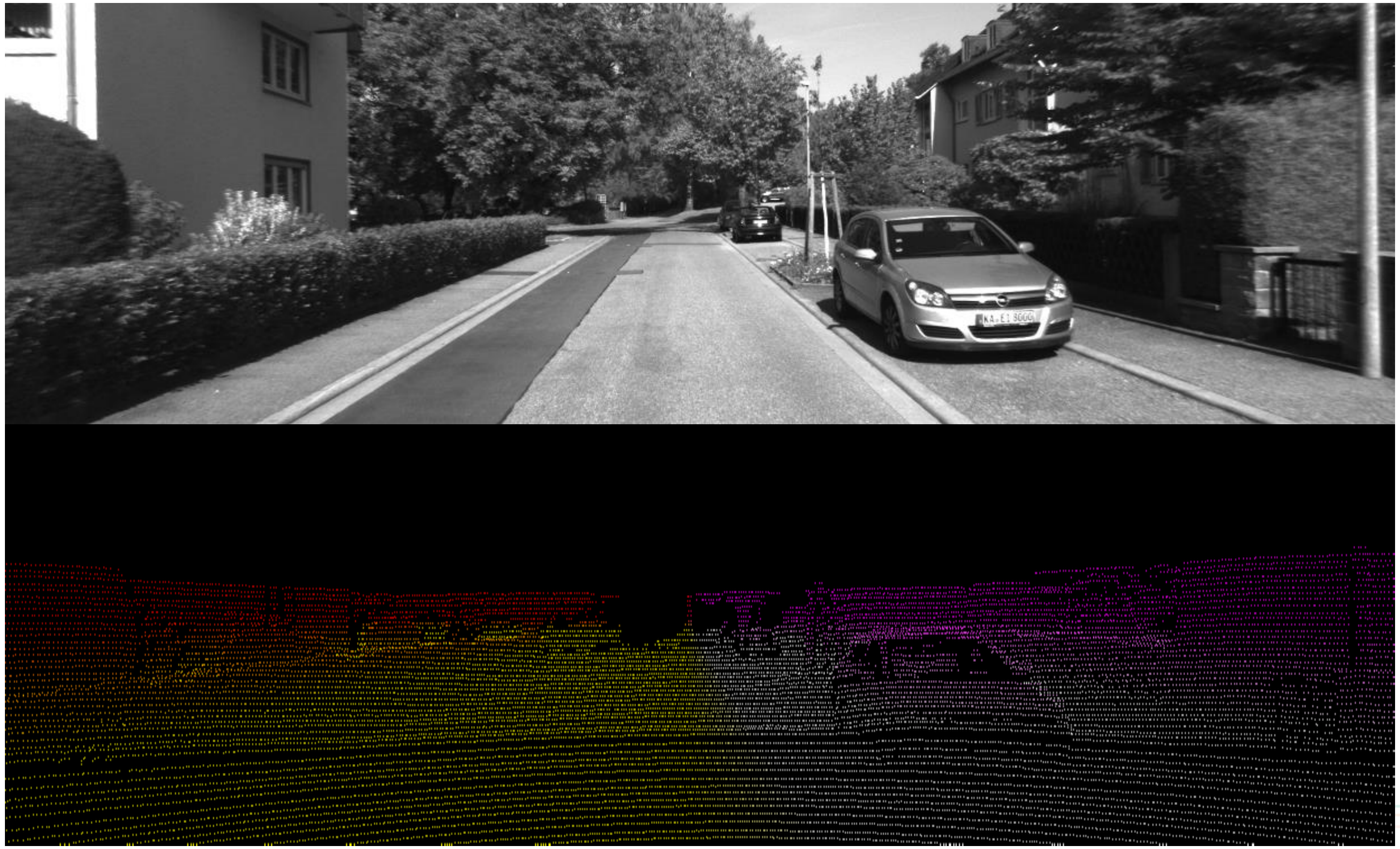
- **3D-2D motion estimation**



$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

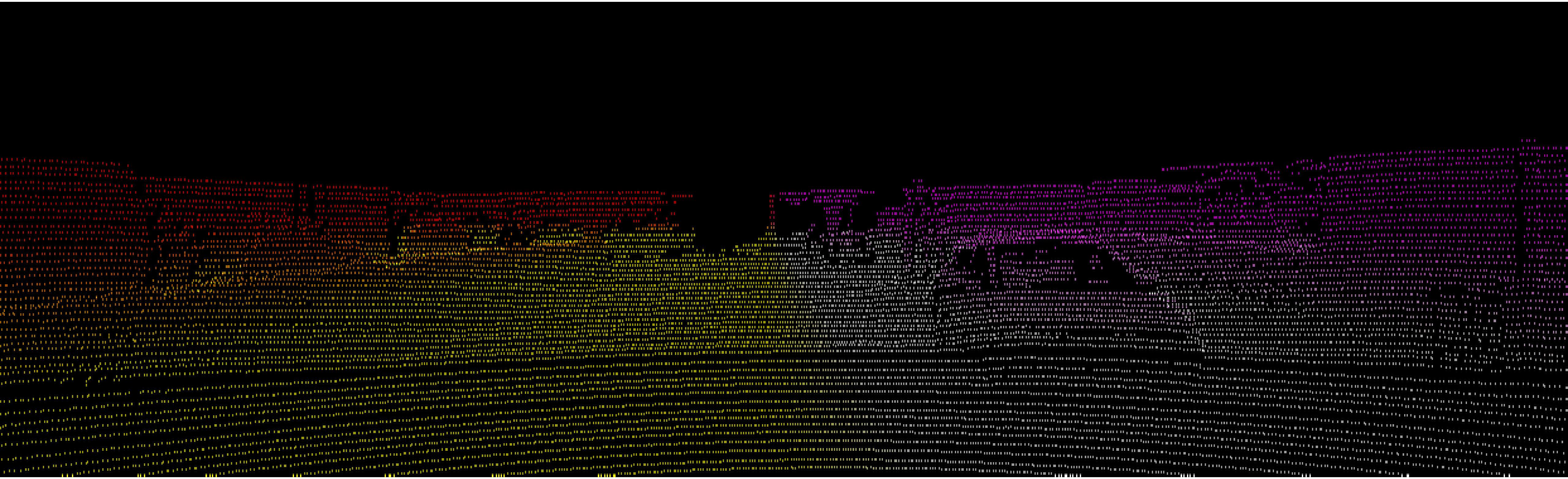
Interpolation

- **3D-2D motion estimation**
  - Sparse LiDAR data



Interpolation

- **3D-2D motion estimation**
  - Sparse LiDAR data

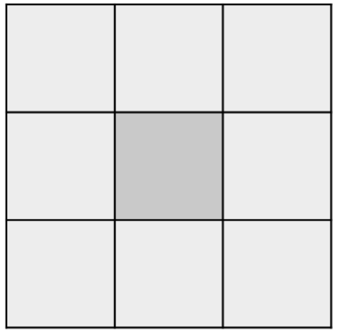


Interpolation

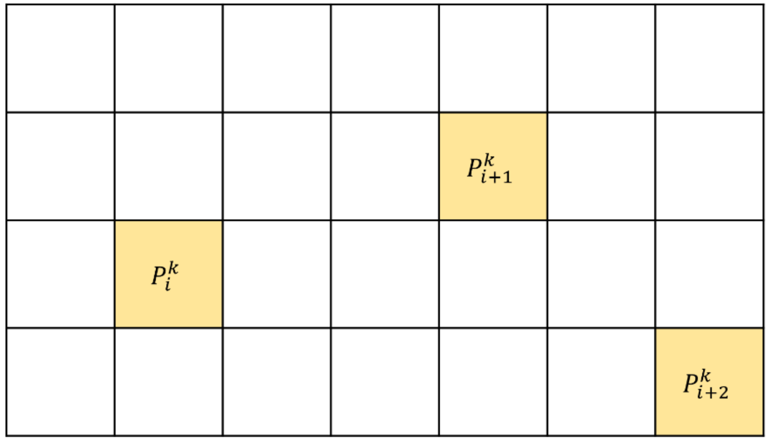
- **Method 1**
  - Dilation interpolation

$$S_j = (p_i^k \oplus B)$$

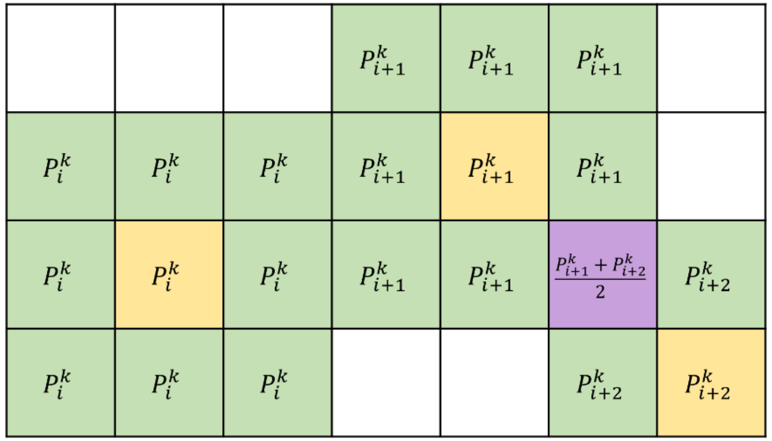
$$S_j \leftarrow P_i^k$$



B



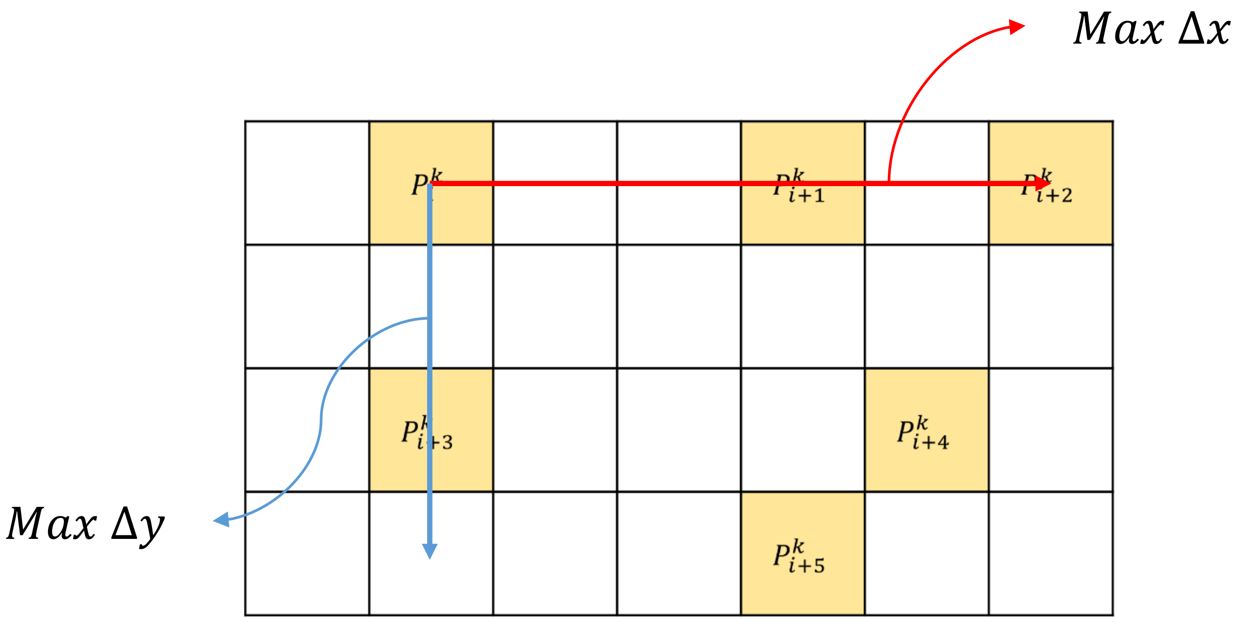
LiDAR image



Interpolated LiDAR image

Interpolation

- **Method 2**
  - Adaptive bilinear interpolation* ($Max\ \Delta x,\ Max\ \Delta y$)



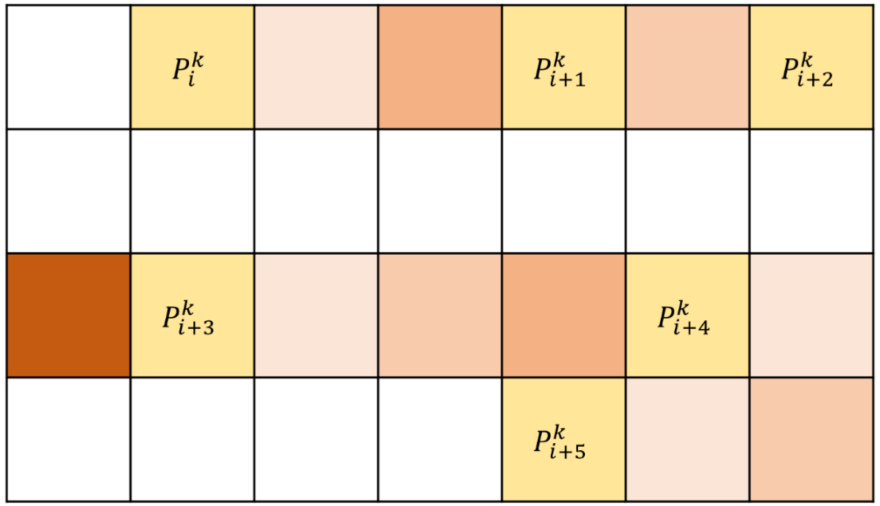
LiDAR image

$$Interpolated\ Point:\ P^k = G * \Delta x + P^k_i$$

$$Gradient:\ G = \frac{P^k_i - P^k_{i+1}}{\Delta x^{i+1}_i}$$

Interpolation

- **Method 2**
  - Adaptive bilinear interpolation* ($Max\ \Delta x,\ Max\ \Delta y$)



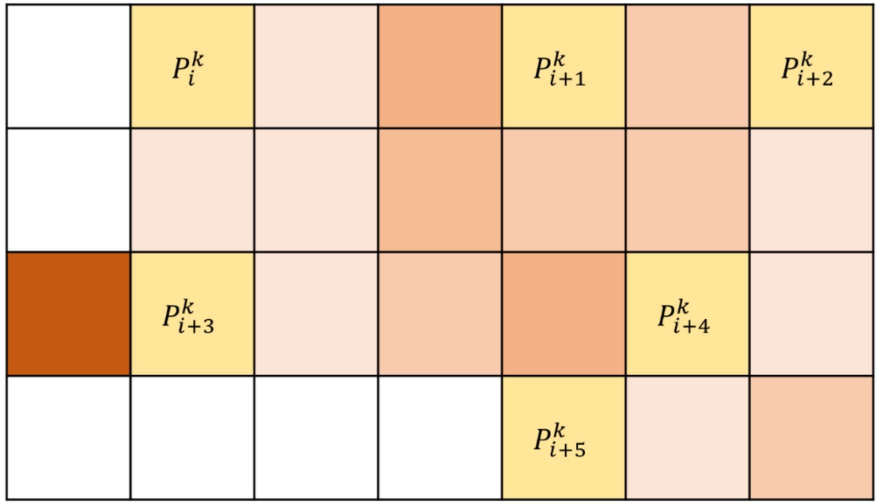
LiDAR image

$$Interpolated\ Point:\ P^k = G * \Delta x + P_i^k$$

$$Gradient:\ G = \frac{P_i^k - P_{i+1}^k}{\Delta x_i^{i+1}}$$

Interpolation

- **Method 2**
  - Adaptive bilinear interpolation* ($Max\ \Delta x,\ Max\ \Delta y$)



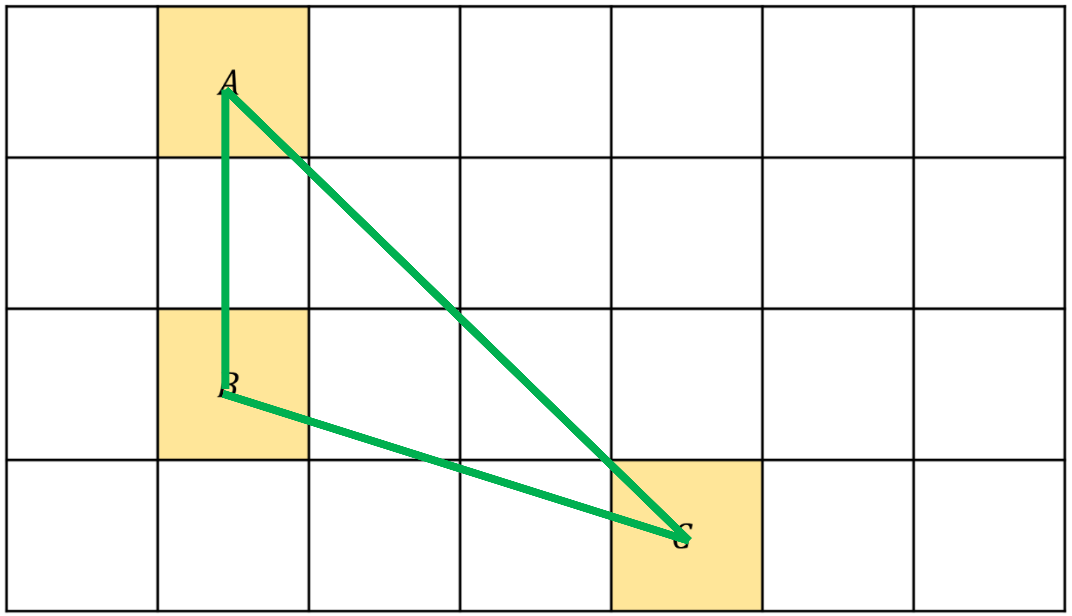
LiDAR image

$$Interpolated\ Point:\ P^k = G * \Delta x + P_i^k$$

$$Gradient:\ G = \frac{P_i^k - P_{i+1}^k}{\Delta x_i^{i+1}}$$

Interpolation

- **Method 3**
  - Plane interpolation

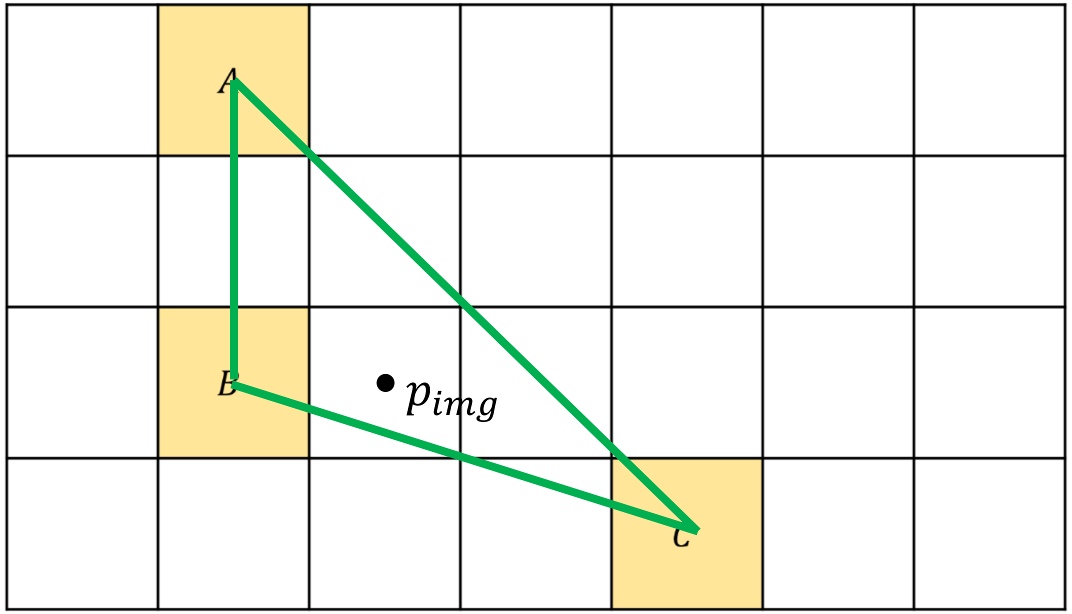


LiDAR image

$$\left.\begin{matrix} A(X_A, Y_A, Z_A) \\ B(X_B, Y_B, Z_B) \\ C(X_C, Y_C, Z_C) \end{matrix}\right\} \quad Plane\ ABC$$

Interpolation

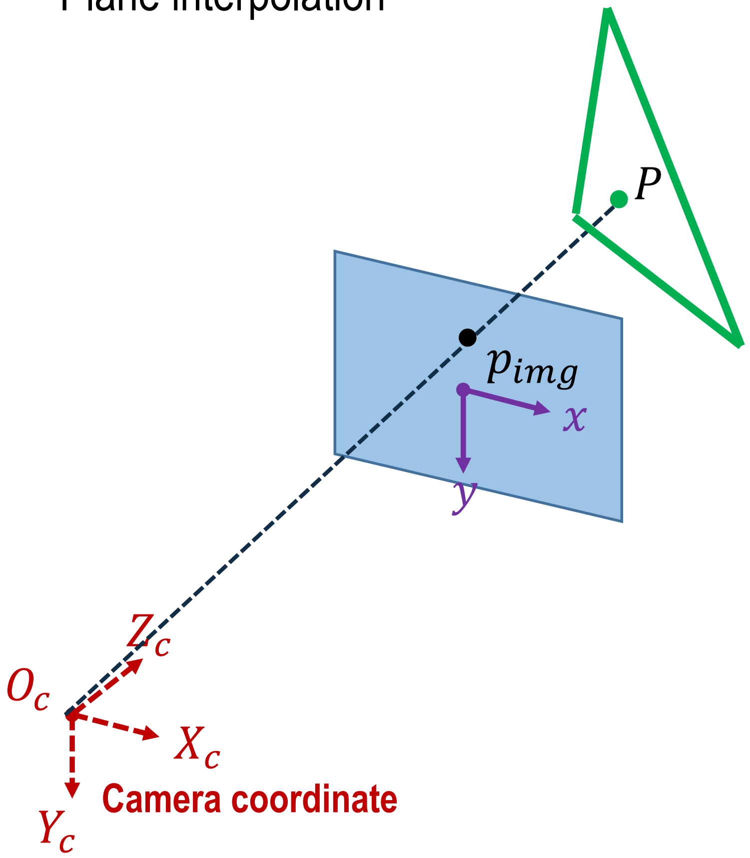
- **Method 3**
  - Plane interpolation



LiDAR image

Interpolation

- **Method 3**
  - Plane interpolation



$$p_{img}2D(x, y)$$

$$\downarrow$$

$$p_{img}2D(x-u, y-v)$$

$$\downarrow$$

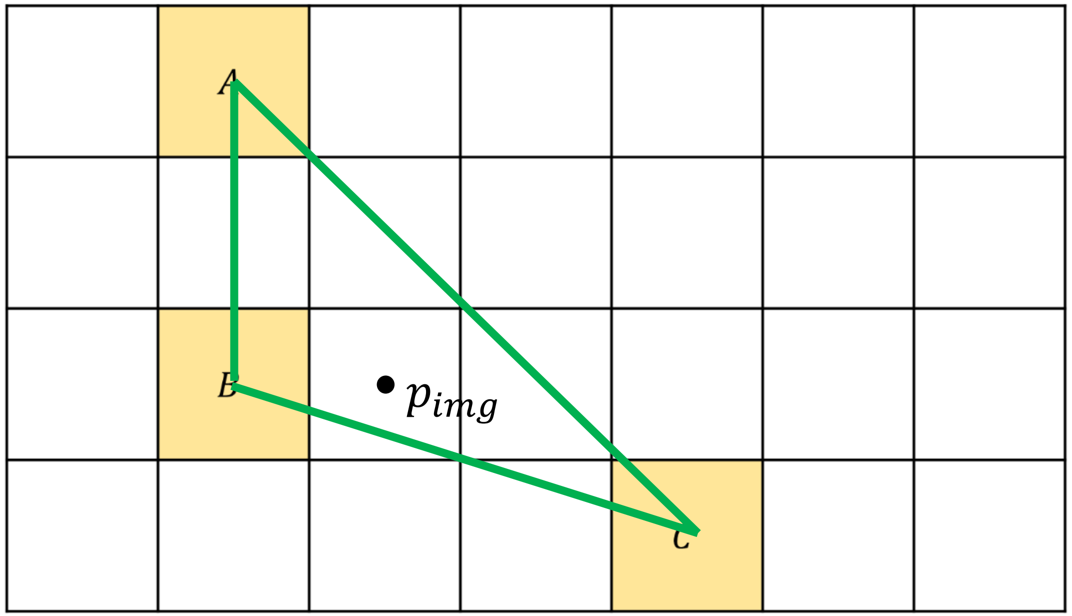
$$p_{img}3D(x-u, y-v, f)$$

$$\downarrow$$

$$line(O_c, p_{img}3D)$$

Interpolation

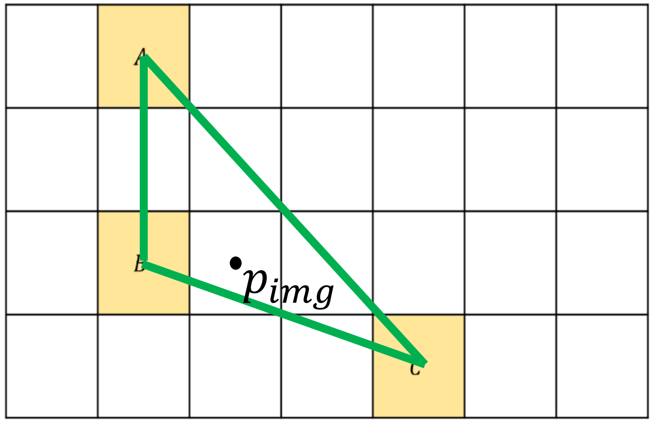
- **Method 3**
  - Plane interpolation



LiDAR image

Interpolation

- **Method 3**
  - Plane interpolation



LiDAR image

$$line\ AB : y = a_{AB} * x + b_{AB}$$

$$line\ BC : y = a_{BC} * x + b_{BC}$$

$$line\ CA : y = a_{CA} * x + b_{CA}$$

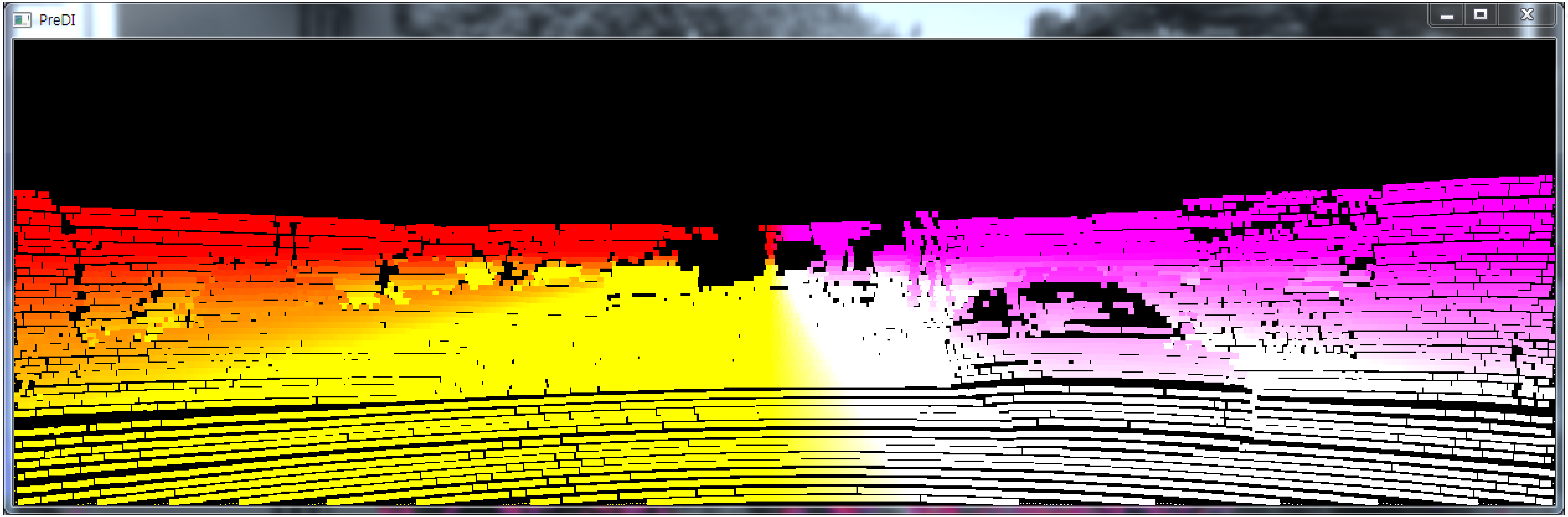
$valid\ function$:

$$if\ ({p_{img}}_y - a_{AB} * {p_{img}}_x + b_{AB}) * \left(C_y - a_{AB} * C_x + b_{AB}\right) < 0\ , continue;$$

$$if\ \ a_{AB} = \infty, \quad \left((A\ or\ B)_x - {p_{img}}_x\right) * \left((A\ or\ B)_x - C_x\right) < 0, continue;$$

Result

- LiDAR data – dilation interpolation

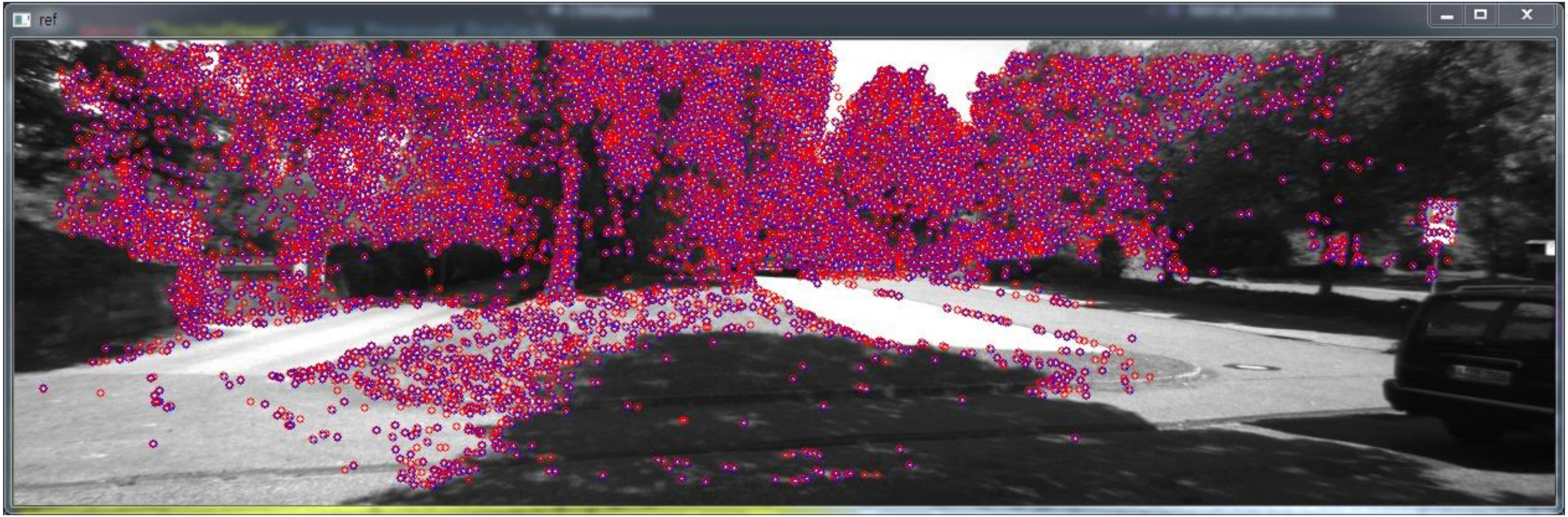


Result

- Reprojection error – dilation interpolation

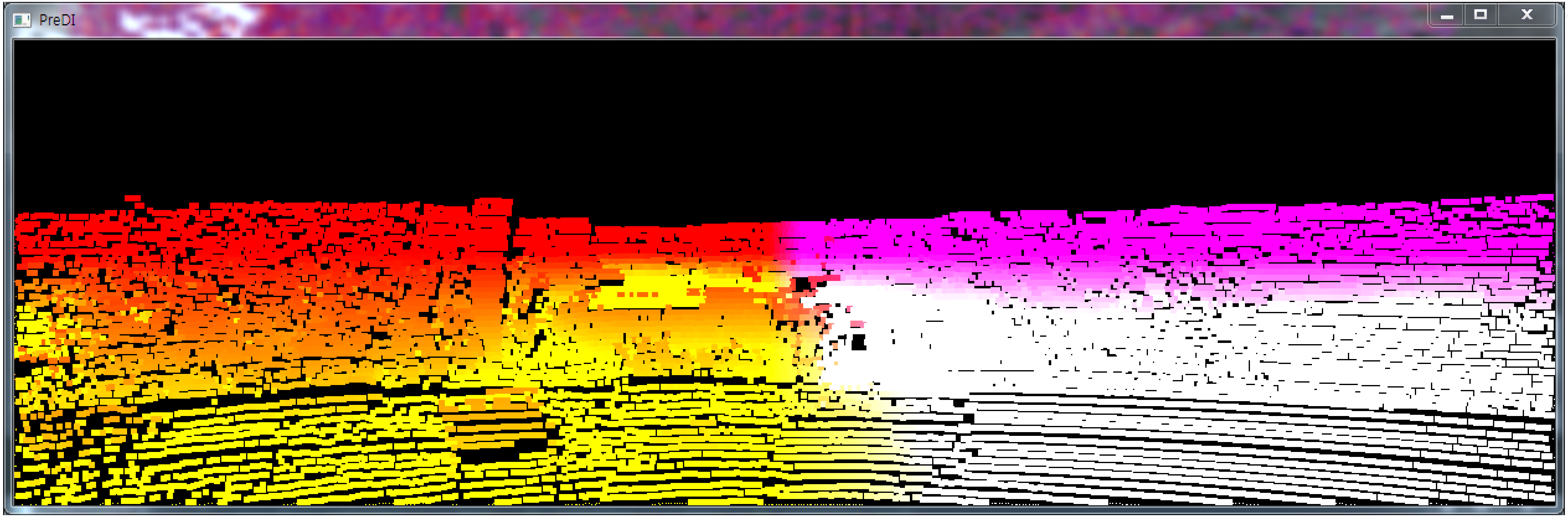
Result

- Reprojection error – dilation interpolation



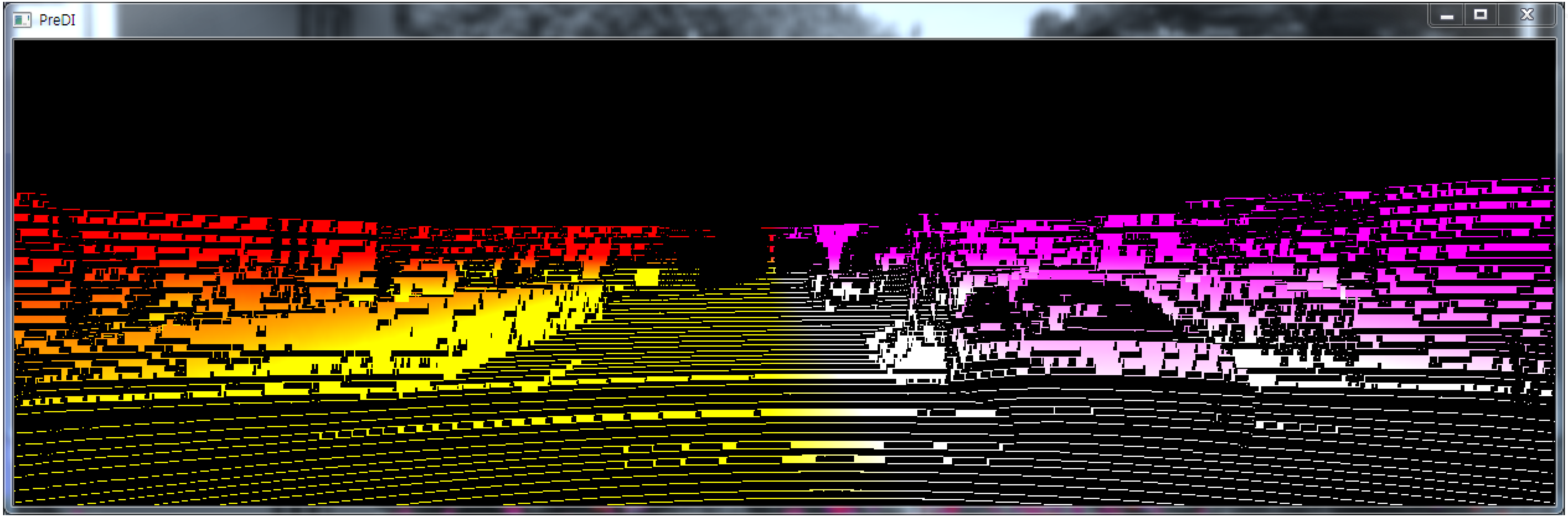
Result

- Outlier



Result

- LiDAR data – adaptive bilinear interpolation

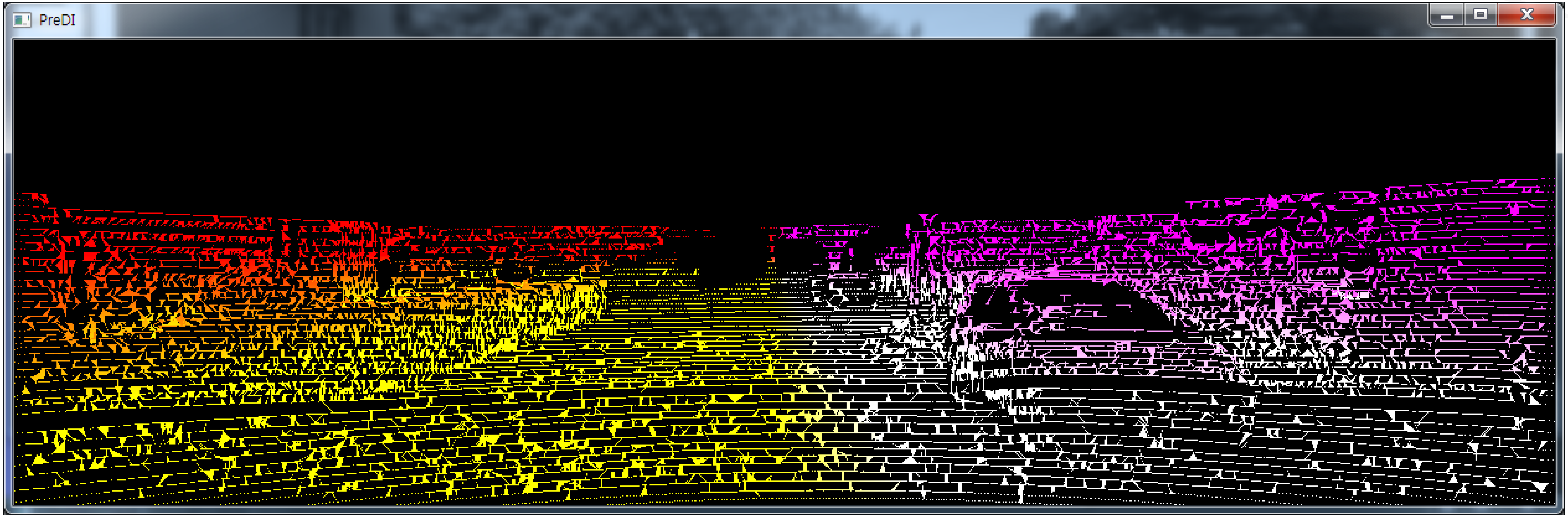


Result

- Reprojection error – adaptive bilinear interpolation

Result

- LiDAR data – plane interpolation

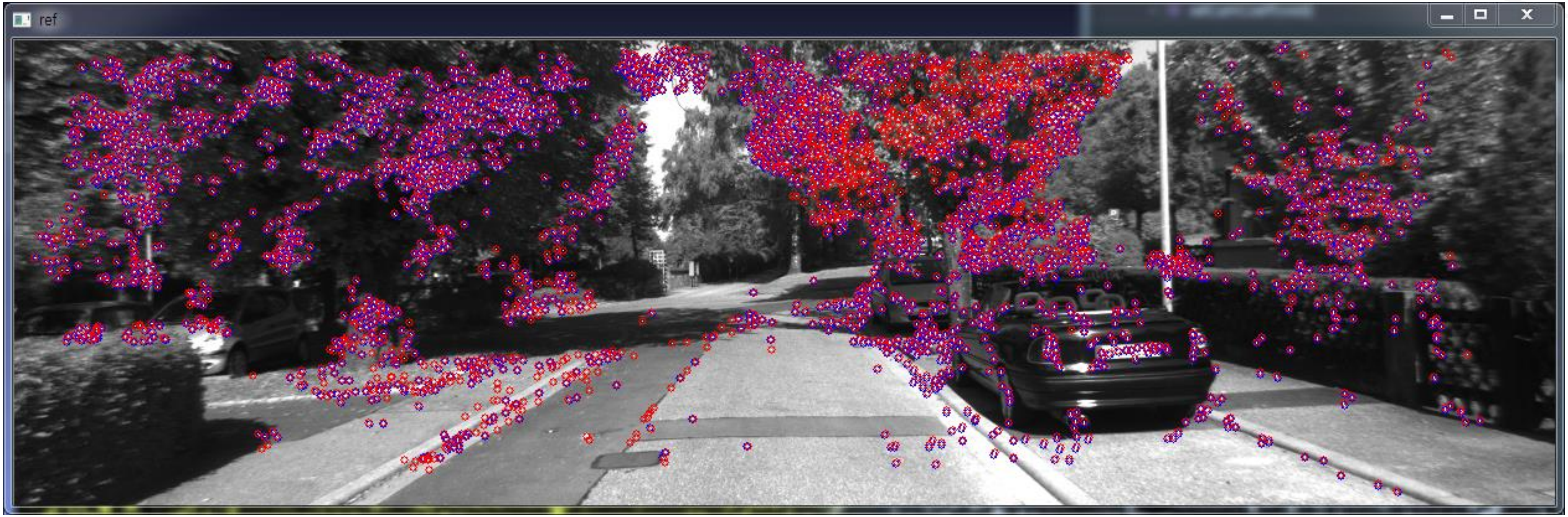


Result

- Reprojection error - plane interpolation

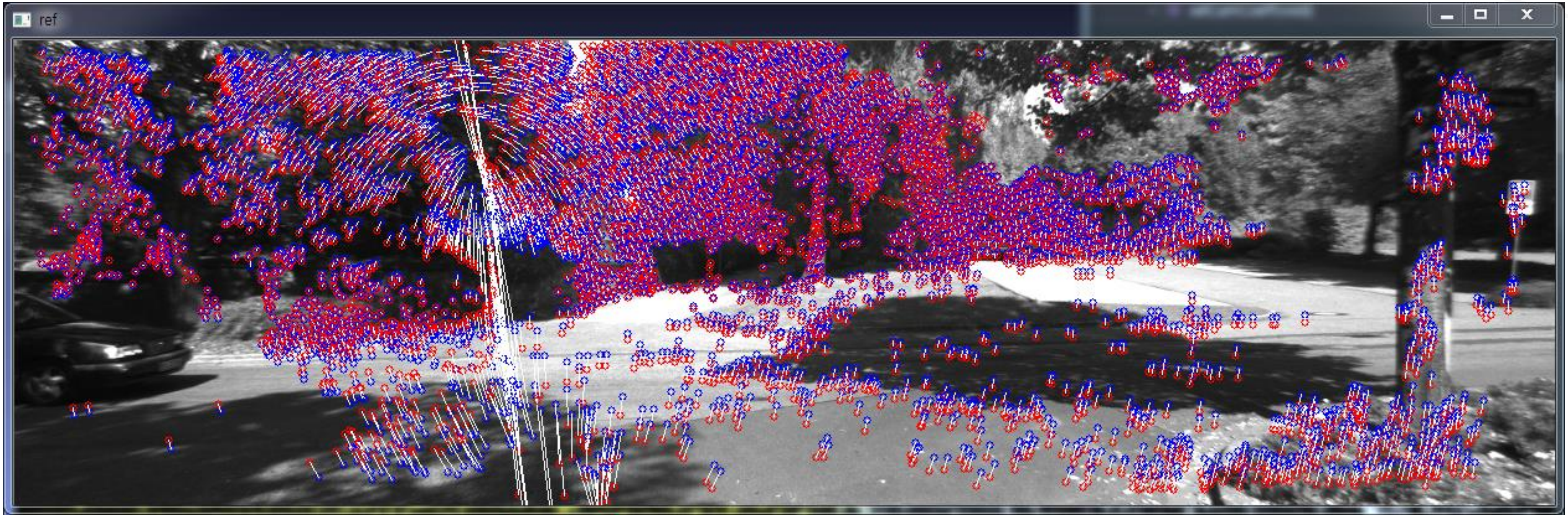
Result

- Reprojection error - plane interpolation



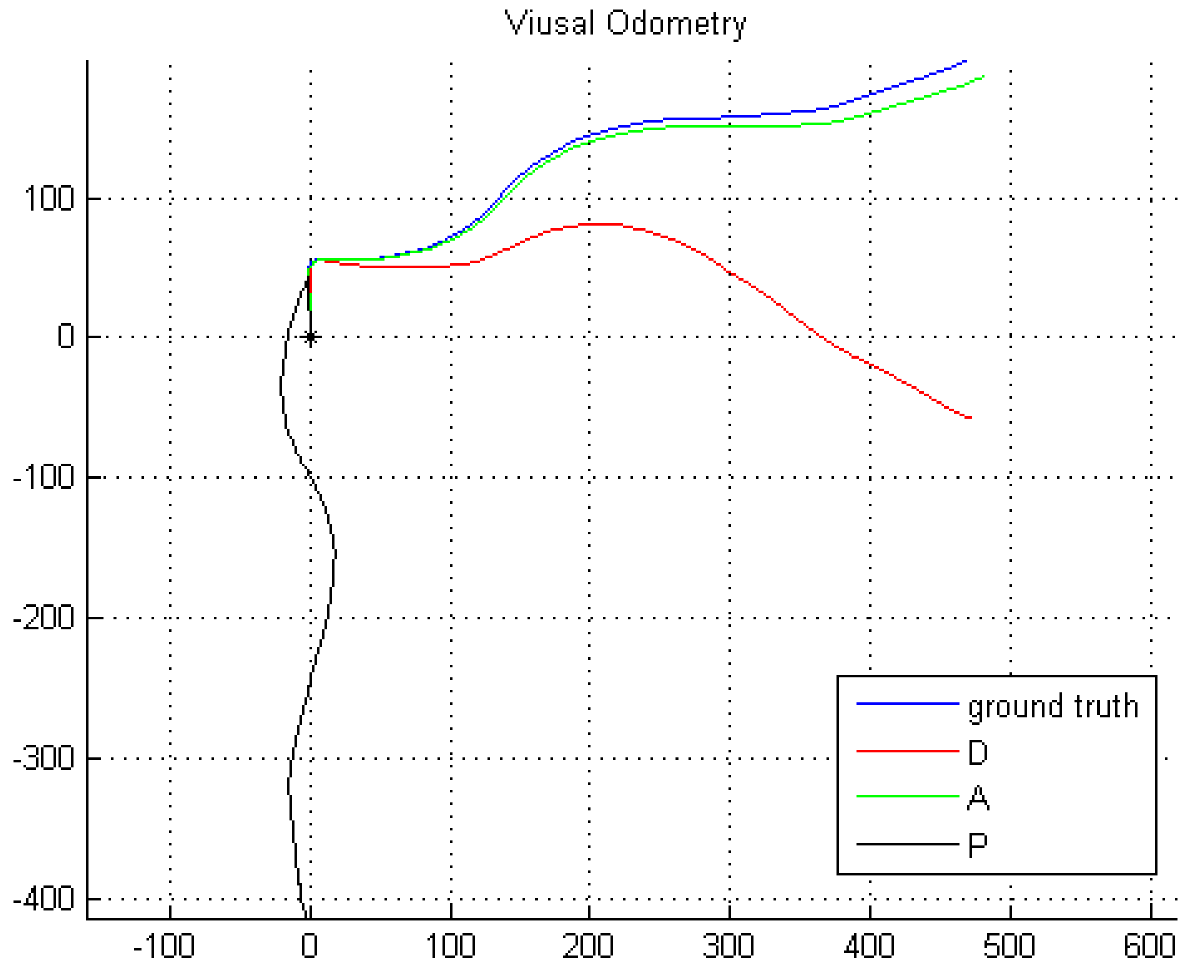
Result

- Reprojection error - plane interpolation



Result

- **Visual Odometry**



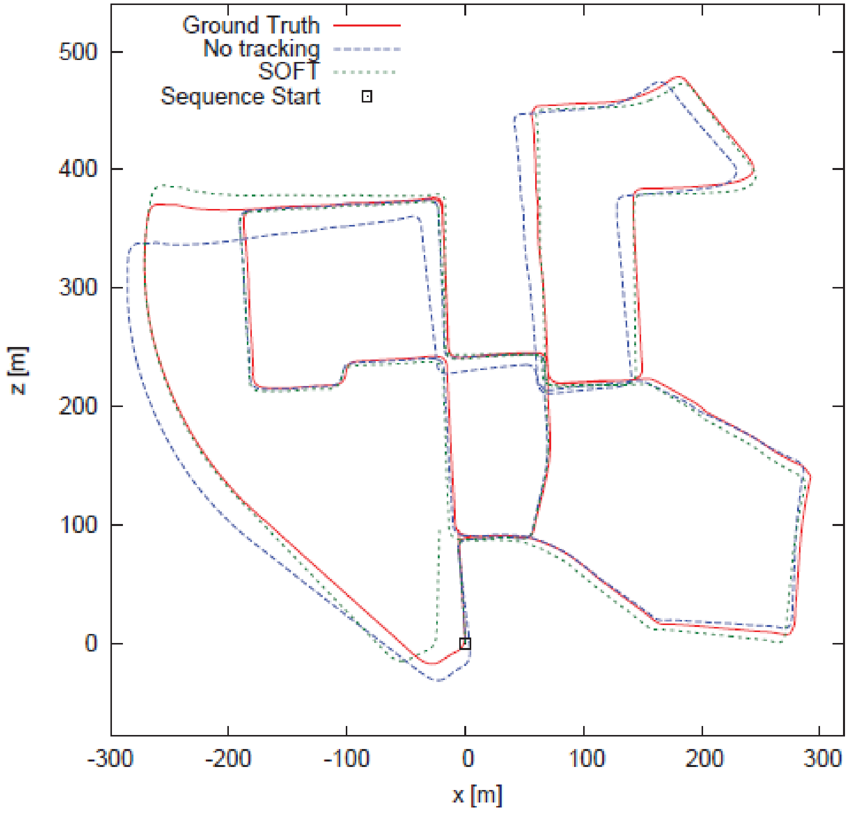
Conclusion

- **SOFT (Steroe Odometry based on careful feature selection and tracking)**

	Method	Setting	Code	Translation	Rotation	Runtime	Environment	Compare
1	V-LOAM			0.68 %	0.0016 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
J. Zhang and S. Singh: Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast. IEEE International Conference on Robotics and Automation(ICRA) 2015.								
2	LOAM			0.70 %	0.0017 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
J. Zhang and S. Singh: LOAM: Lidar Odometry and Mapping in Real-time. Robotics: Science and Systems Conference (RSS) 2014.								
3	SOFT2			0.81 %	0.0022 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
4	GDVO			0.86 %	0.0031 [deg/m]	0.09 s	1 core @ >3.5 Ghz (C/C++)	
5	HypERROCC			0.88 %	0.0027 [deg/m]	0.25 s	2 cores @ 2.0 Ghz (C/C++)	
6	SOFT			0.88 %	0.0022 [deg/m]	0.1 s	2 cores @ 2.5 Ghz (C/C++)	
I. Cvišić and I. Petrović: Stereo odometry based on careful feature selection and tracking. European Conference on Mobile Robots (ECMR) 2015.								

Conclusion

- SOFT (Steroe Odometry based on careful feature selection and tracking)



Reconstructed path of the KITTI00 dataset

Stereo camera with IMU

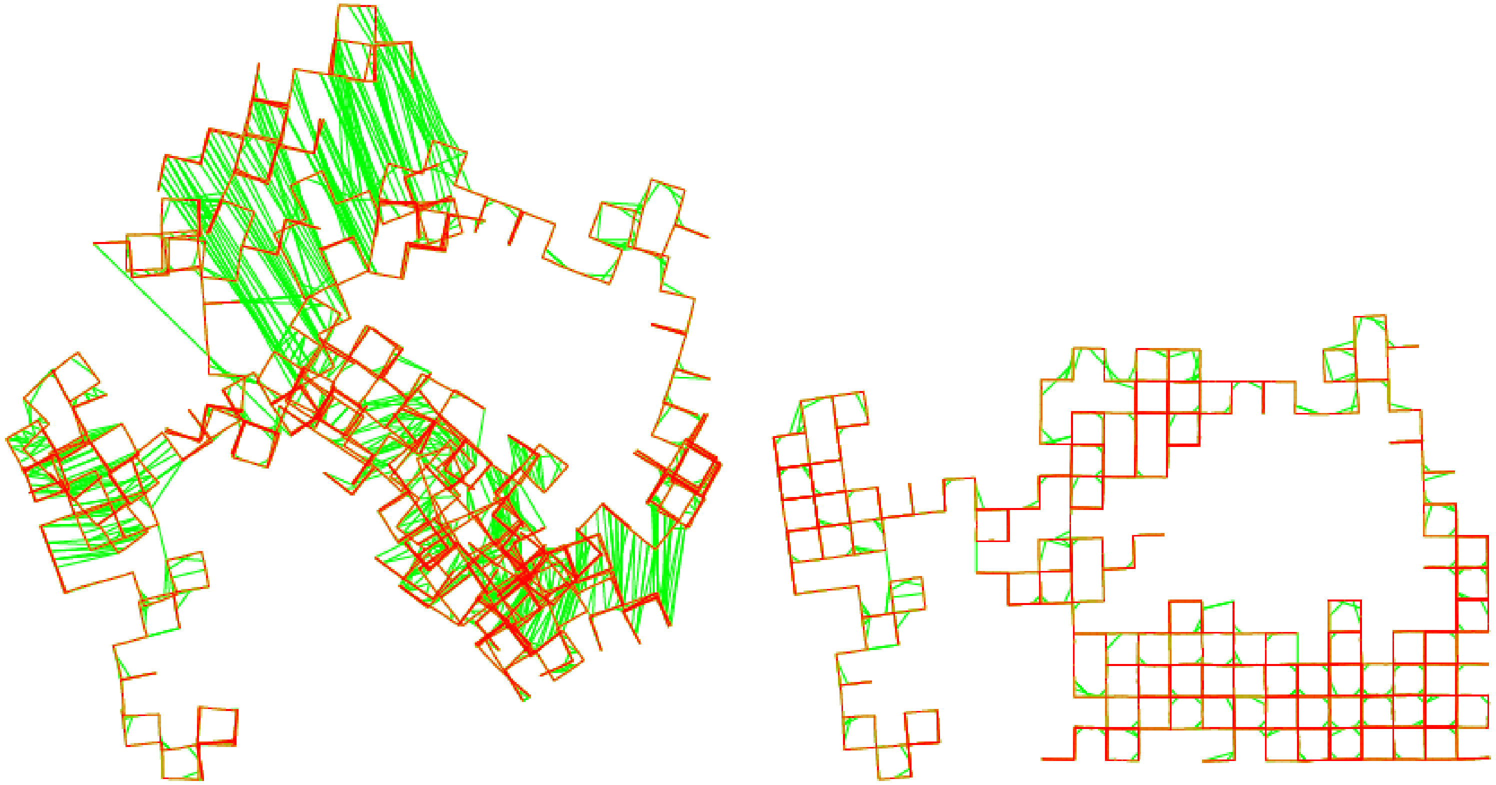
Conclusion

- SOFT (Steroe Odometry based on careful feature selection and tracking)

Path estimated from self recorded dataset

Conclusion

- Optimization – iSAM (Inceremetal Smoothing and Mapping)



(a) Original noisy data set. (b) Trajectory after incremental optimization.

Conclusion

- Optimization – iSAM (Inceremetal Smoothing and Mapping)

(a) Trajectory based on odometry only.

(b) Trajectory and map after incremental optimizia-tion.

Q&A