ELTE | FACULTY OF INFORMATICS

PointNet & CalibRRNet

Massinissa Aouragh:
Faculty of Informatics, Department of Artificial Intelligence
Robert Bosch Kft
m2j7au@inf.elte.hu

Deep Learning on PointCloud

- End to end networks for classification, semantic segmentation and object detection since 2017
- Registration and odometry
- Initial applications on transformed point cloud (voxel grids, multi view images projection)

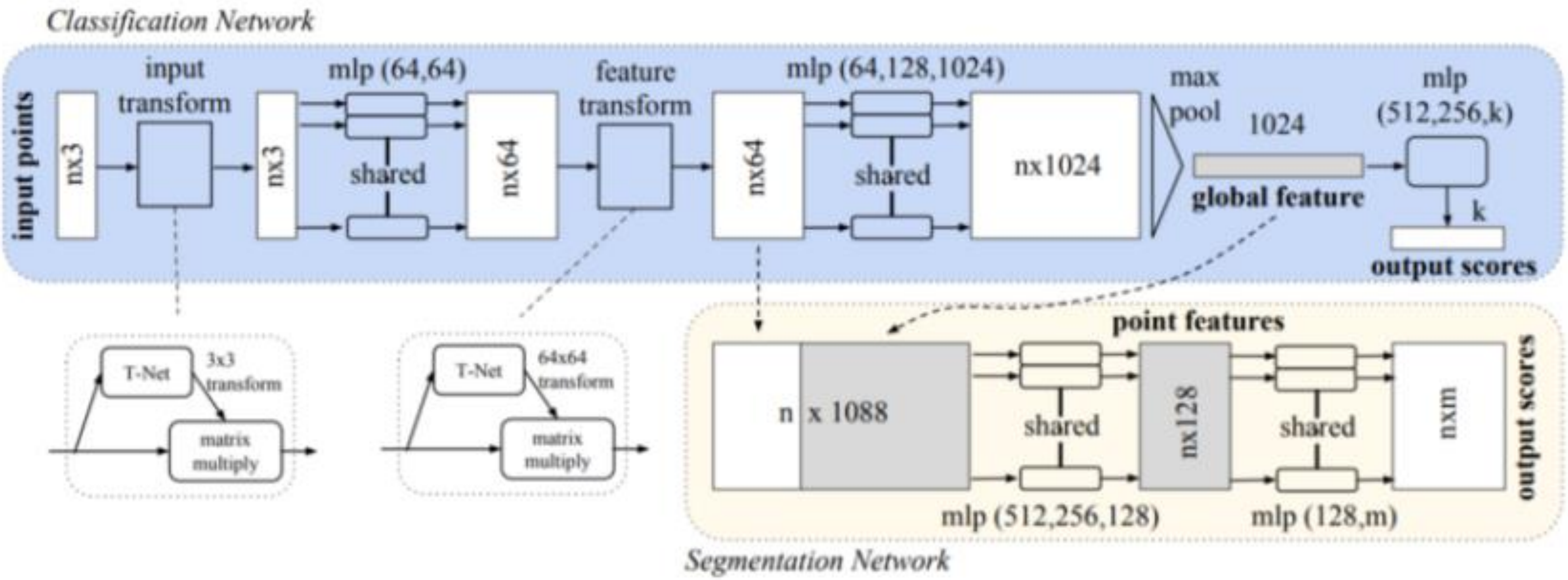
PointNet

- First work to employ Deep learning directly for processing 3D point clouds
- Considers the unordered nature of points, interactions of points in a local region and invariance to certain geometric transformations
- Network should be invariant to N! permutations
- Network should infer correctly under rotation, translation and affine transformation

PointNet

- To solve the order invariance, the network uses max pooling
- The Network applies first an input transform using a T-Net, to ensure invariance to geometric transformations
- Second the transformed points are passed through a sequence of pointwise multilayer perceptron for higher dimensional feature space
- A feature transform is then applied with the same purpose to make the features invariant to transformations
- Point Features are aggregated using max pooling to form a global feature vector

Point Net



Classification Network
input points
nx3
input transform
nx3
mlp (64,64)
shared
nx64
feature transform
nx64
mlp (64,128,1024)
shared
nx1024
max pool
1024
global feature
mlp (512,256,k)
k
output scores
T-Net
3x3 transform
matrix multiply
T-Net
64x64 transform
matrix multiply
point features
n x 1088
shared
nx128
shared
nxm
output scores
mlp (512,256,128)
mlp (128,m)
Segmentation Network

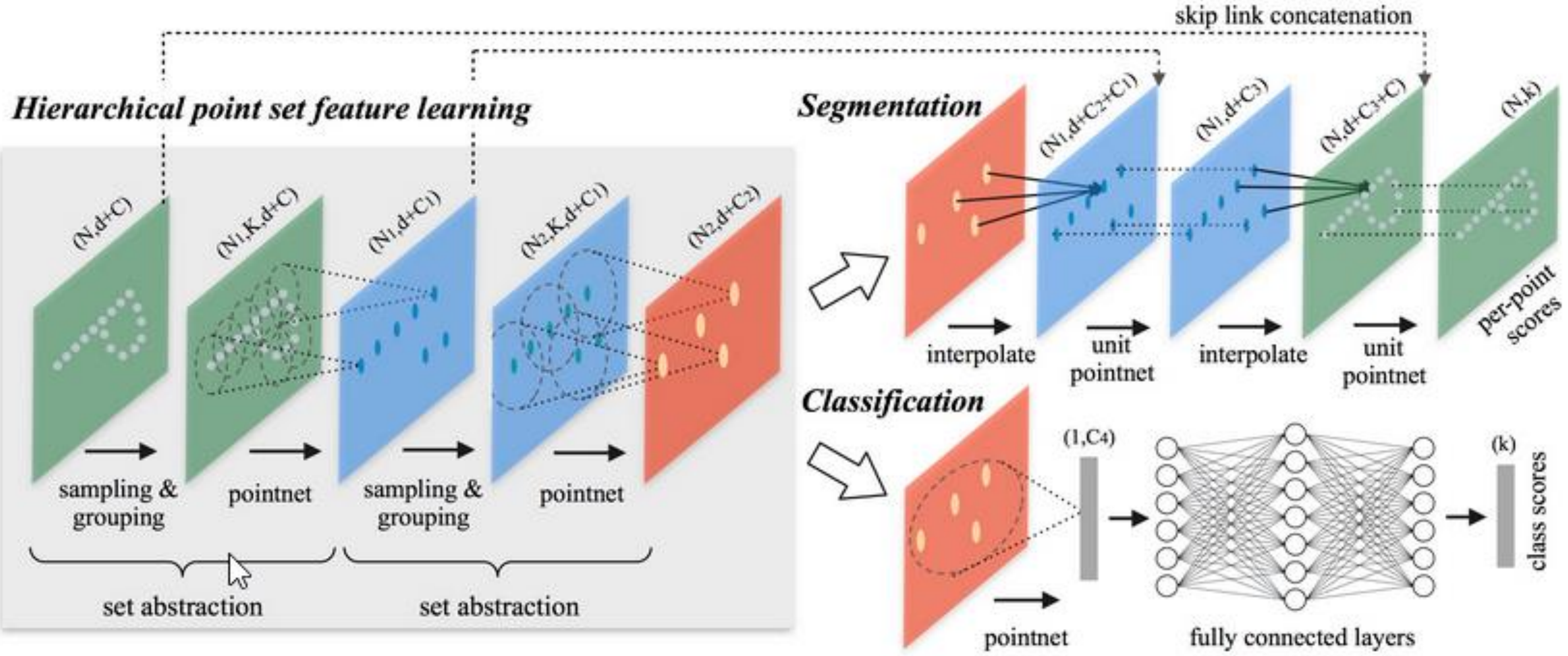
PointNet++

- PointNet does not capture information about the local context of points at different scales
- PointNet++ proposes a hierarchical feature learning framework to address PointNet limitations
- The hierarchical learning process is achieved by a series of set abstraction levels
- Each abstraction level consist of sampling layer, grouping layer and PointNet Layer

PointNet++

- In the sampling layer a subset of m points $\{x_{i1}, x_{i2}, \ldots, x_{im}\}$ is sampled from the n points. The iterative farthest point sampling technique is used. The sampled m points form the set of centroids for the grouping layer
- Grouping layer takes the input point set N*(D+C), and the coordinates of the centroids N*D from the sampling step. All points laying inside a sphere of a certain radius around each centroid are collected forming set N'*K*(D+C)
- PointNet is applied for each set of group points after being translated to a local system centered at the centroid

PointNet++



skip link concatenation
Hierarchical point set feature learning
Segmentation
Classification
(N,d+C)
(N1,K,d+C)
(N1,d+C1)
(N2,K,d+C1)
(N2,d+C2)
sampling & grouping
pointnet
sampling & grouping
pointnet
set abstraction
set abstraction
(N1,d+C2+C1)
(N1,d+C3)
(N,d+C3+C)
(N,k)
interpolate
unit pointnet
interpolate
unit pointnet
per-point scores
(1,C4)
(k)
pointnet
fully connected layers
class scores

Calibration Network

Target based calibration:

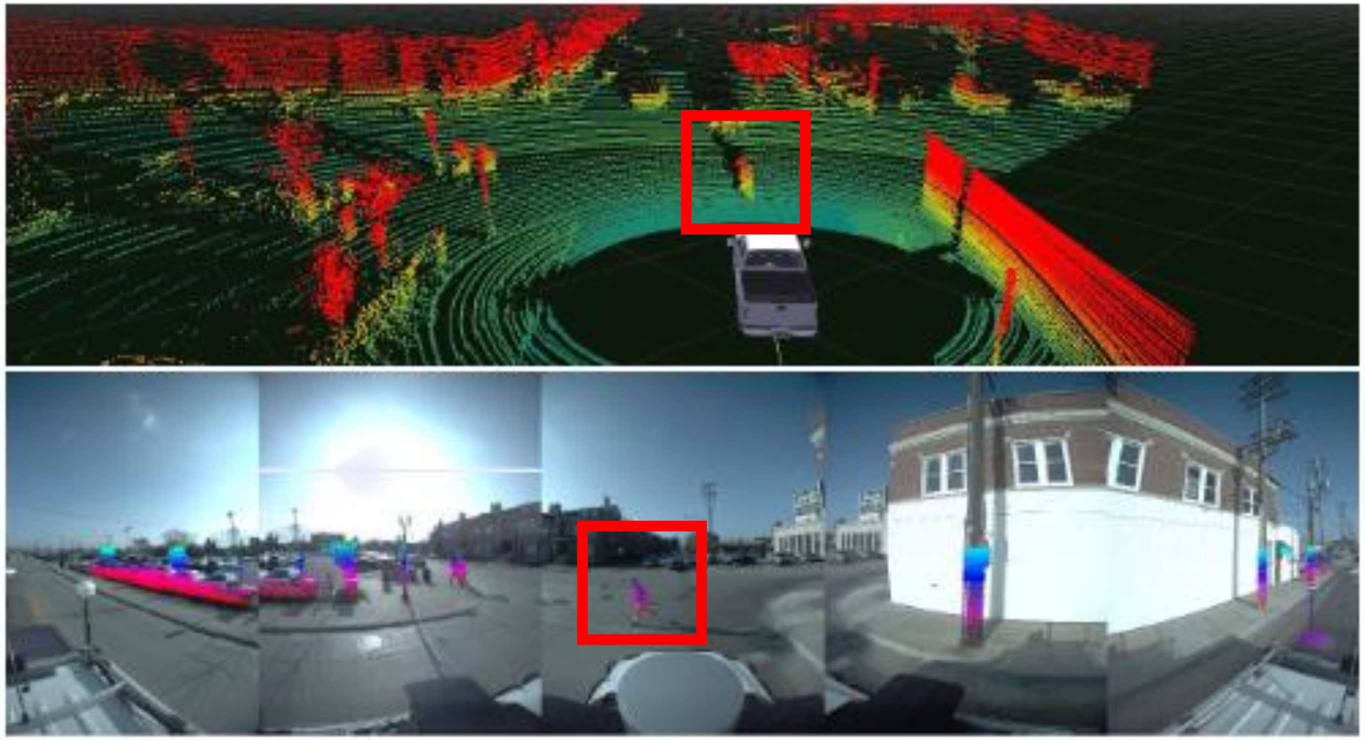
- Dedicated calibration target
- A priori knowledge of target dimension

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8265264
Zoltan Pusztai, Levente Hajder

Target and Targetless calibration

Targetless calibration:

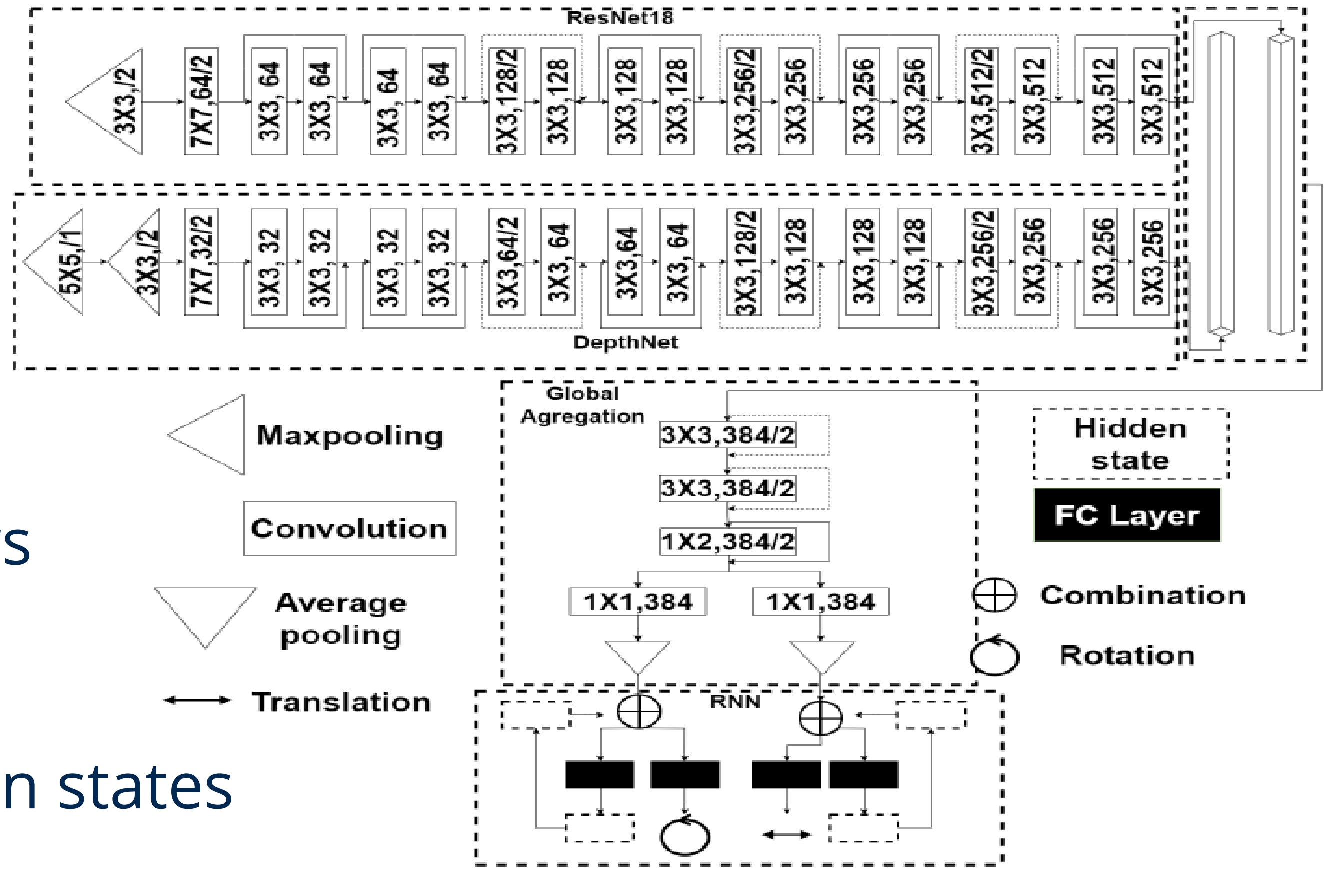
- Laser reflectivity and pixel intensity matching
- Motion sensor dependency



Source: http://robots.engin.umich.edu/publications/gpandey-2012a.pdf
Gaurav Pandey et al.

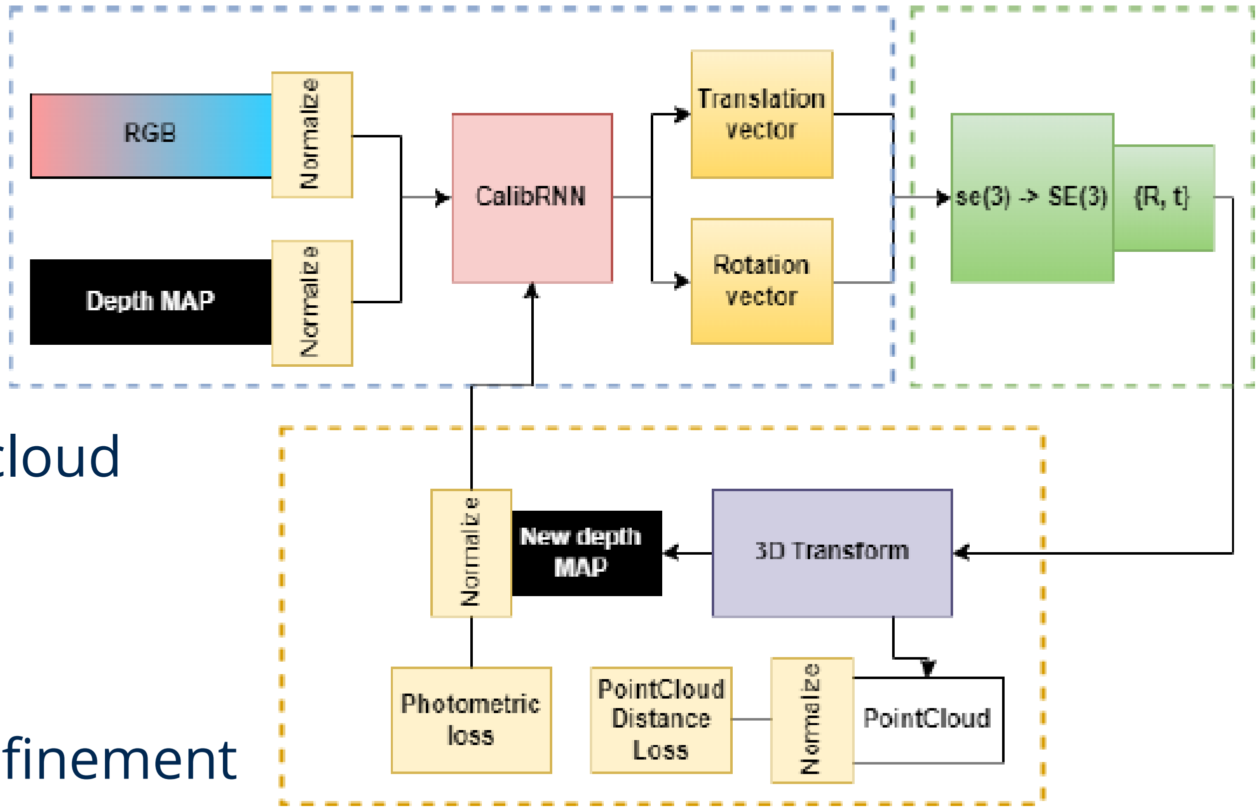
Deep neural networks

- CNN residual networks
- RGB & Depth local special feature extraction
- Feature aggregation
- Estimation of velocity vectors (translation & rotation)
- Translation & rotation hidden states



Framework structure

- RGB & Depth map input
- Translation & rotation estimation
- Mapping from the algebra se(3) to manifold SE(3)
- 6 DoF transformation on the point cloud
- Point distance calculation
- Pixel intensity calculation
- Feedback the new depth map for refinement



Equivalence representations

Yaw Pitch Roll to Mat

$$\mathbf{R}_z(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{R}_y(\chi) = \begin{pmatrix} \cos\chi & 0 & \sin\chi \\ 0 & 1 & 0 \\ -\sin\chi & 0 & \cos\chi \end{pmatrix}$$

$$\mathbf{R}_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix}$$

Quat to Mat

$$\begin{matrix} q_r^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_r q_z) & 2(q_z q_x + q_r q_y) \\ 2(q_x q_y + q_r q_z) & q_r^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_r q_x) \\ 2(q_z q_x - q_r q_y) & 2(q_y q_z + q_r q_x) & q_r^2 - q_x^2 - q_y^2 + q_z^2 \end{matrix}$$

Applied Lie Algebra

se(3) -> SE(3):
Rodrigues' Formula:

$$e^{\mathbf{v}} \equiv e^{\mathbf{A}(\mathbf{v})} = \begin{pmatrix} e^{\boldsymbol{\omega}^\wedge} & \mathbf{Vt} \\ 0 & 1 \end{pmatrix}$$

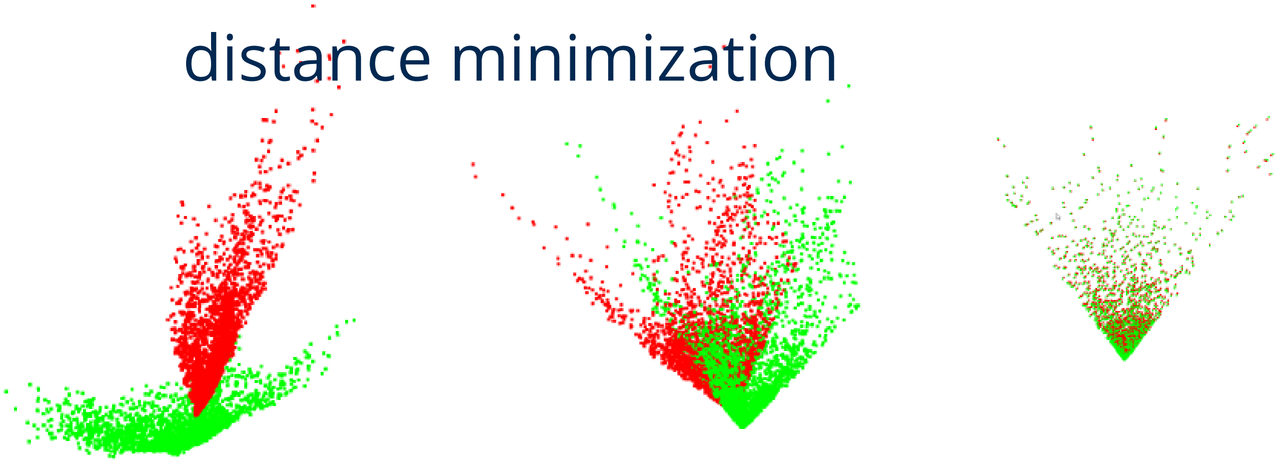
$$\mathbf{V} = \mathbf{I_3} + \frac{1-\cos\theta}{\theta^2}\omega^\wedge + \frac{\theta - \sin\theta}{\theta^3}(\omega^\wedge)^2$$

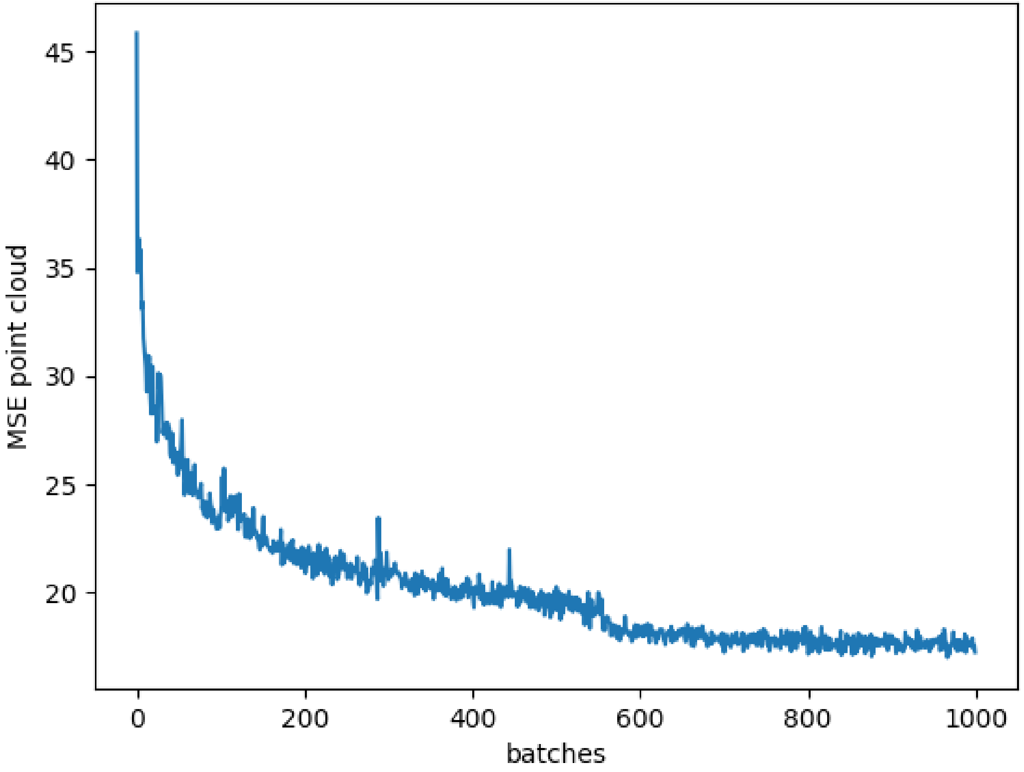
$$e^{\boldsymbol{\omega}} \equiv \text{matexp}(\omega^\wedge) = \mathbf{I_3} + \frac{\sin\theta}{\theta}\omega^\wedge + \frac{1-\cos\theta}{\theta^2}(\omega^\wedge)^2$$

$$\omega = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \omega^\wedge = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}$$

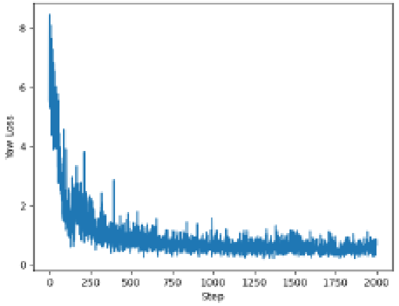
Training

- Geometrically supervised learning
- Dynamic training dataset generation
- Highest focus on point cloud distance minimization

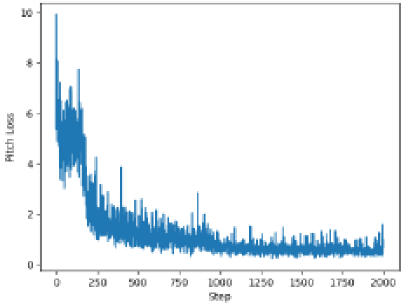




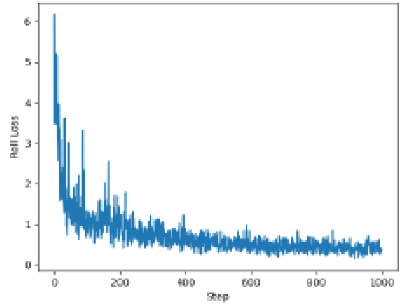
Reported Errors



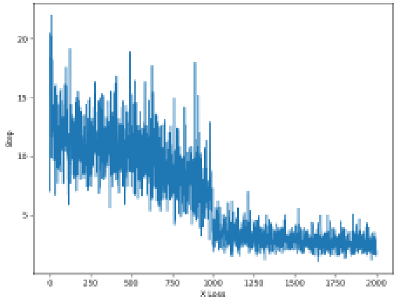
Yaw angle loss in °



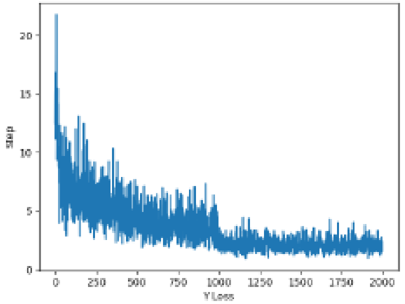
Pitch angle loss in °



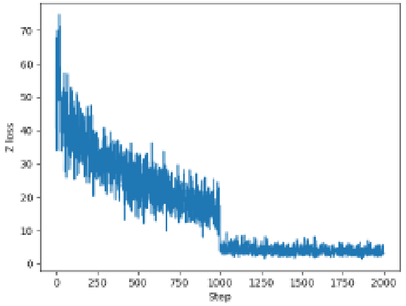
Roll angle loss in °



X loss in cm



Y loss in cm



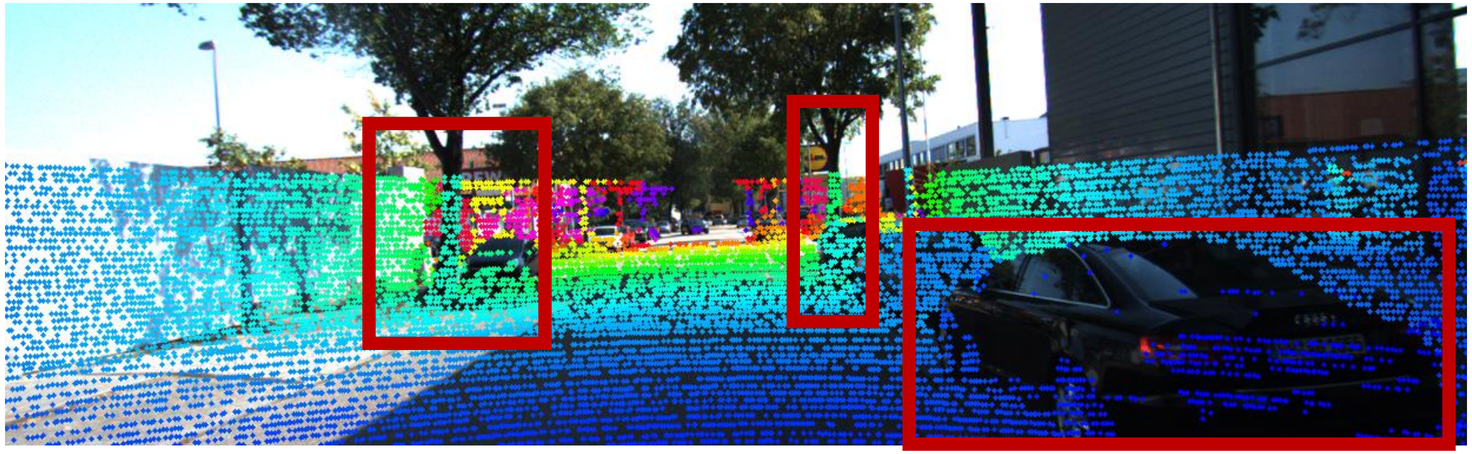
Z loss in cm

Reported Errors

Approach	Yaw°	Pitch°	Roll°	X com	Y cm	Z cm
Regnet	0.24	0.25	0.36	7	7	4
Calibnet	**0.15**	0.9	0.18	12.1	3.49	7.87
CalibRRnet	0.2043	**0.2290**	**0.0931**	**1.0590**	**0.9319**	**1.4131**

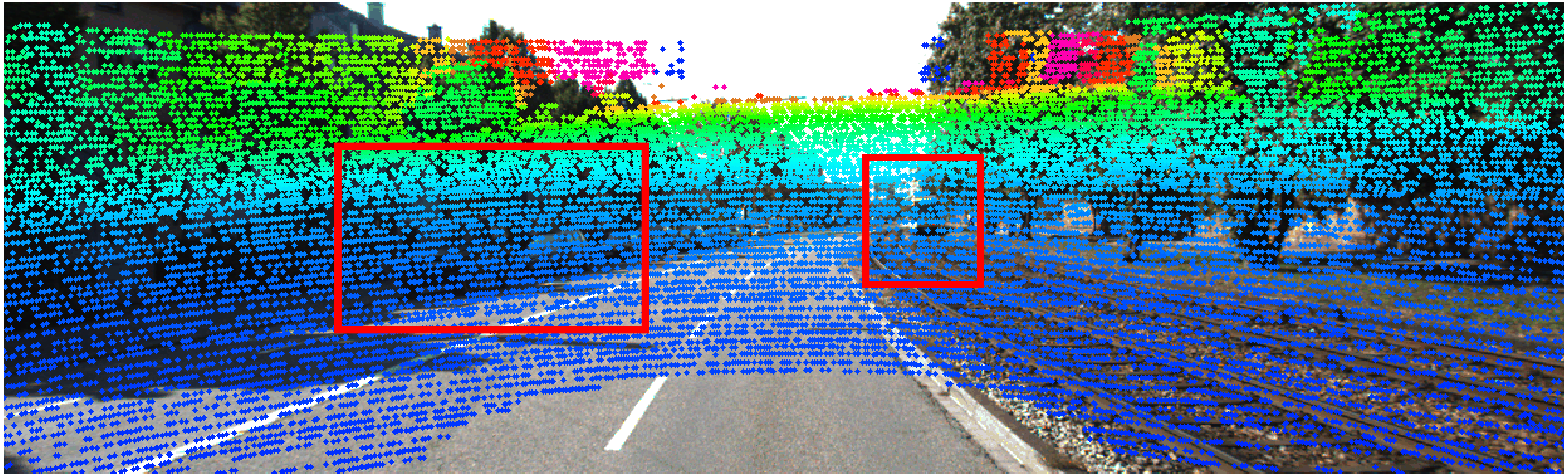
Calibration

- Misaligned lidar point cloud projection

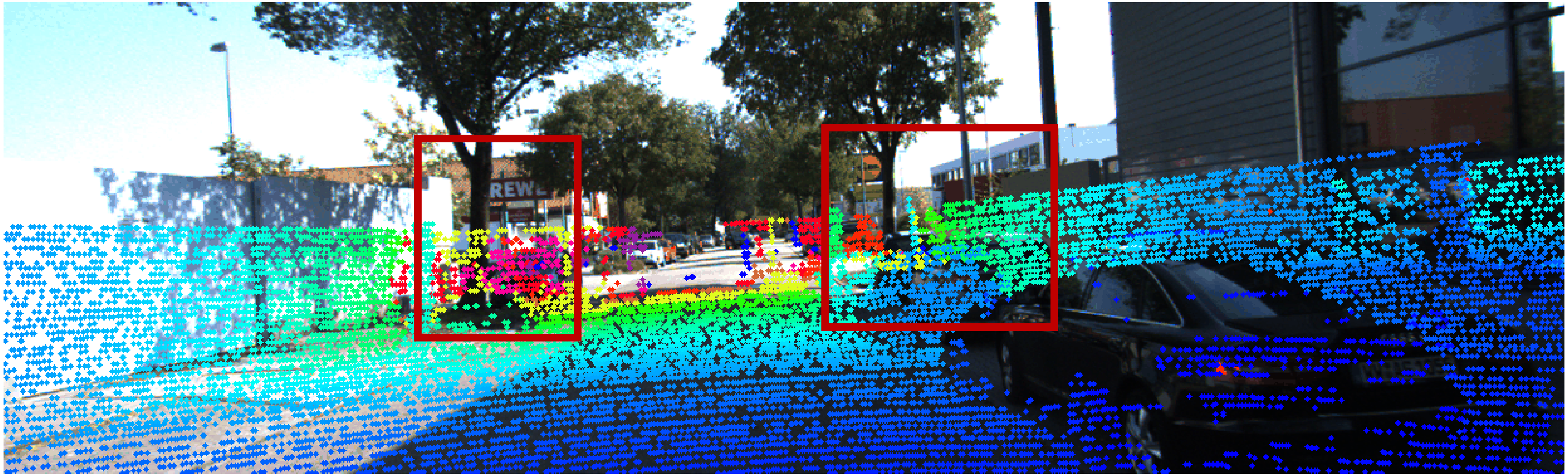


- Alignment result of Depth map feature and RGB feature map.

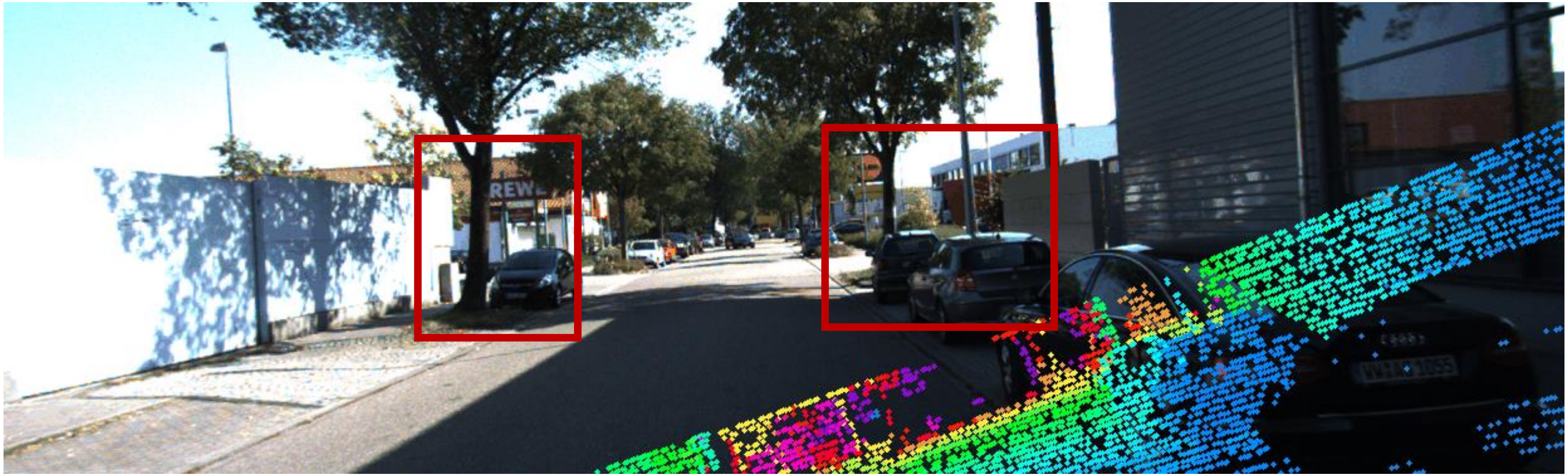
Rotation(deg): [-10, 10], Translation(m): [-0.2, 0.2]



Rotation(deg): [-10, 10], Translation(m): [-0.2, 0.2]

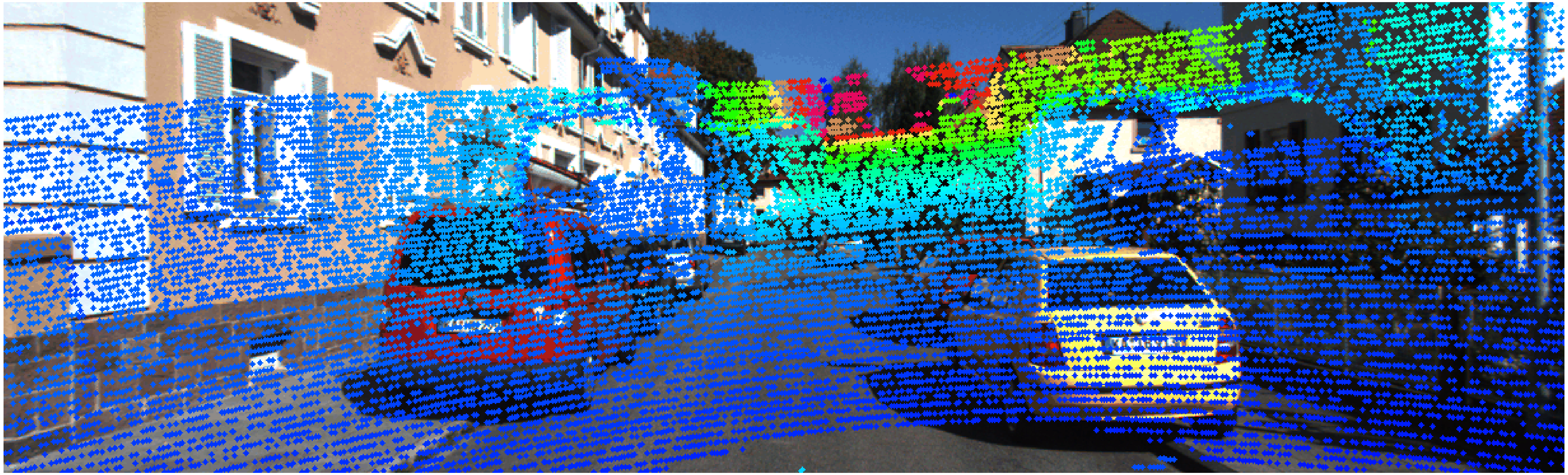


Rotation(deg): [-20, 20], Translation(m): [-0.4, 0.4]



Model application on different camera calibration setup

Rotation(deg): [-10, 10], Translation(m): [-0.2, 0.2]



Model application on different camera calibration setup

Rotation(deg): [-20, 20], Translation(m): [-0.4, 0.4]