AlignNet-3D

Fast Point Cloud Registration of Partially Observed Objects

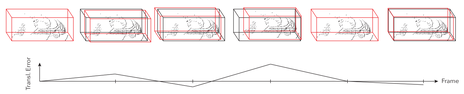
Johannes Groß, Aljoša Ošep, Bastian Leibe (Computer Vision Group, RWTH Aachen)

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Code and data

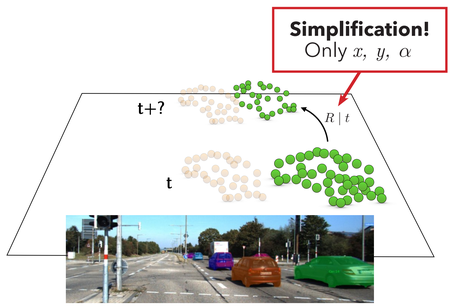
Motivation

While the majority of trackers focus on data association, precise state (3D pose) estimation is often only coarsely estimated by approximating targets with centroids or (3D) bounding boxes.

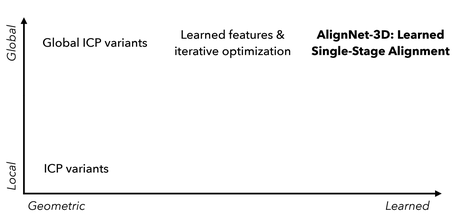


In automotive scenarios, motion perception of surrounding agents is critical and inaccuracies in the vehicle close-range can have catastrophic consequences. Instead of approximating targets with their centroids, our approach is capable of utilizing noisy 3D point segments of objects to estimate their motion.

The task of **Precise 3D Tracking** is, given an existing tracker (providing tracklets with identity associations), similar to 3D point cloud registration. We simplify the general 3D registration task to predicting a restricted rigid transform on the ground plane $(x,\ y,\ \alpha)$.

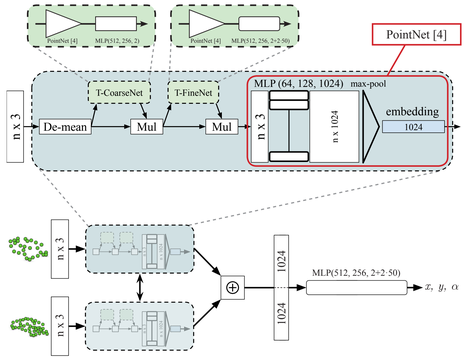


Existing methods for point cloud registration include variants of Iterative Closest Points (ICP) [1], which often get stuck in local minima. Global registration methods improve on this, but are too slow for most multi-object tracking applications. With AlignNet-3D, we aim to predict good alignments with a fast, end-to-end trainable one-stage prediction.



Method

Network



- Input to the network are the two point cloud tracklets
- Two siamese branches bring them into canonical poses, for which an embedding is then computed
  - **T-CoarseNet** predicts a canonical center of the de-meaned input point cloud
  - **T-FineNet** predicts another canonical center and canonical orientation of the transformed point cloud
  - Both sub-networks are supervised with the ground truth pose of the input point cloud
- The final transform prediction is supervised by the remaining ground truth transform

Training

- Input point cloud size n: 512
- Color channel on KITTI is not used
- Batch size: 128

Loss

- T-CoarseNet, T-FineNet and the final transform prediction are all fully supervised
- Translational and angular losses are penalized separately, as in [5]
- Huber loss on translation deviation
- Angle prediction: 50 angle bins + 50 residuals
- Angle loss: $L_{\text{angle}} = L_{\text{cls}} + \lambda \cdot L_{\text{reg}}$

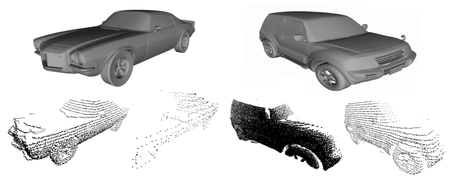
Huber loss on residual to ground truth angle bin

Cross entropy loss on angle class prediciton

Experiments

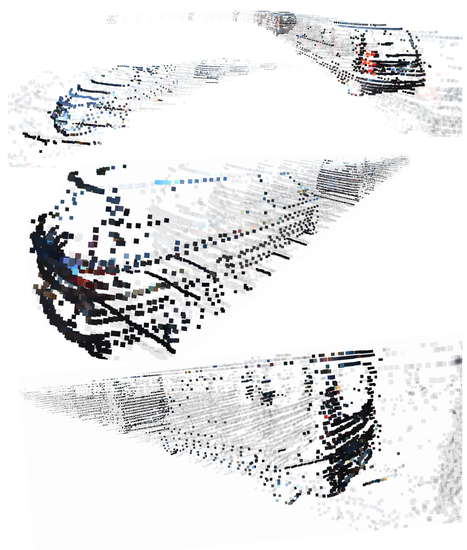
Synthetic Data

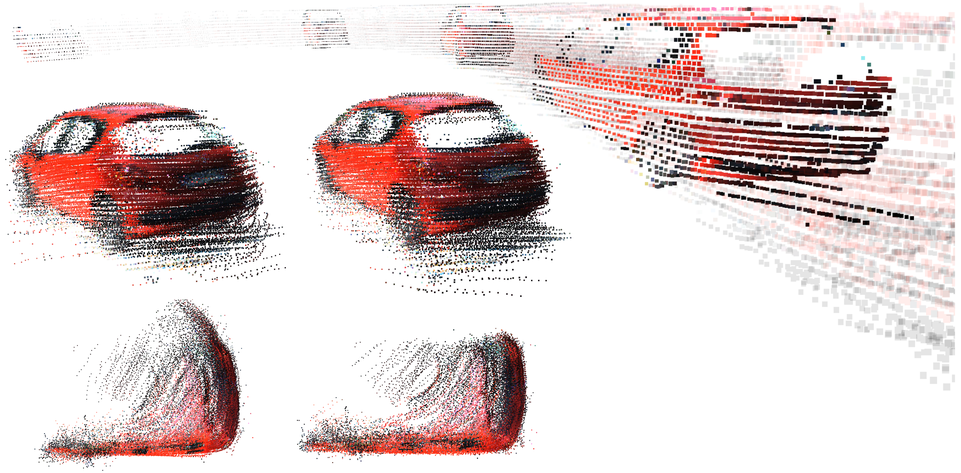
- 8000/1000/1000 scenes (train/val/test)
- Simulated Velodyne 64E laser scanner (same as in KITTI) with CAD models of ModelNet [7]
- Each scene: Two objects at 2-80m distance
- Up to 1m translation and 90° relative rotation
- Gaussian Noise proportional to distance
- Variants: `SynthCars`, `SynthCarsPersons`, `Synth20`, `Synth20others`



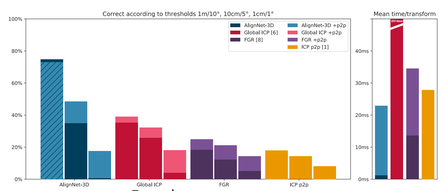
KITTI Tracking Data

- `KITTICars`: 20518/7432/1358 scenes of frame-to-frame car pairs of KITTI Tracking [2]
- Ground truth transform and tracklets are obtained from annotated 3D bounding boxes
- `KITTICarsPersons`: 28463/10003/2069 scenes of car and pedestrian
- `KITTI[...]Hard`: More challenging variants with at least 10 frames distance and 45° angle difference

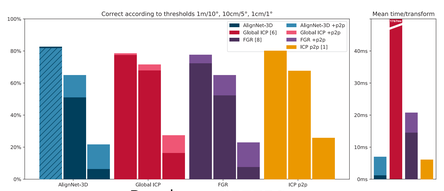




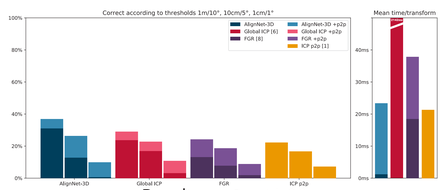
Shown on top are tracklets obtained from a KITTI tracking [2] trajectory (4 of the 35 point clouds are highlighted for illustrative purposes). Below, all tracklets were brought to the same reference frame via frame-to-frame alignments (left: accumulated error, right: non-accumulated error)



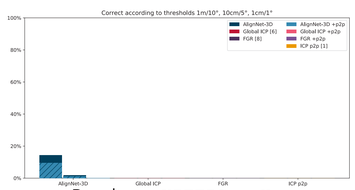
Results on `SynthCars`



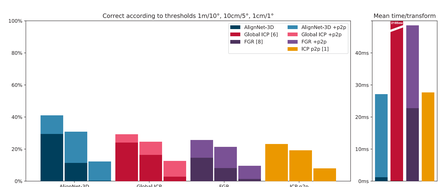
Results on `KITTICars`



Results on `Synth20`



Results on `KITTICarsHard`



Results on `Synth20others`

	d. < 80m	d. < 20m	d. < 5m	
	RMSE v	RMSE v	RMSE v	time/transf.
Centr. Kalman Filter	2.568m/s	1.251m/s	1.604m/s	**0.004ms**
ADH 2D [3]	2.691m/s	1.139m/s	1.023m/s	0.253ms
ADH 2D (parallel)	2.691m/s	1.139m/s	1.023m/s	0.066ms
ADH 3D [3]	2.682m/s	1.132m/s	1.002m/s	0.418ms
AlignNet-3D	**1.834m/s**	**0.851m/s**	**0.732m/s**	1.220ms

Comparison to sequence level trackers Centroid-KF and ADH [3]

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. PAMI, 14(2):239–256, 1992.

[2] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In CVPR, 2012.

[3] D. Held, J. Levinson, S. Thrun, and S. Savarese. Combining 3d shape, color, and motion for robust anytime tracking. In RSS, 2014.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. In CVPR, 2017.

[5] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3D object detection from RGB-D data. CVPR, 2018.

[6] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3d registration. In ICRA, 2009.

[7] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D shapenets: A deep representation for volumetric shapes. In CVPR, 2015.

[8] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In ECCV, 2016.