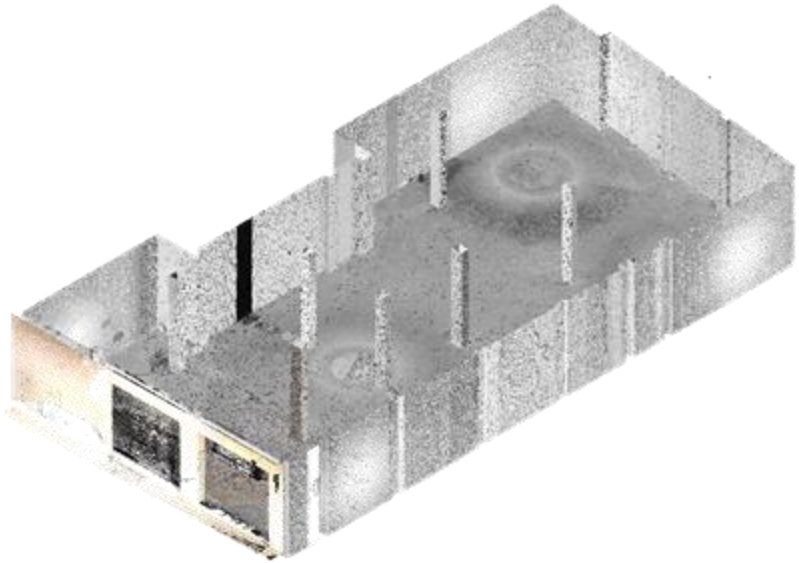
Object Tracking in Dynamic 3D Point Clouds

Hugo Saison

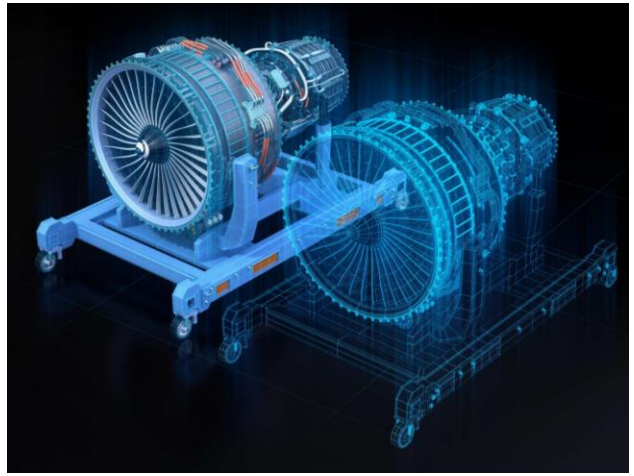
Supervised by

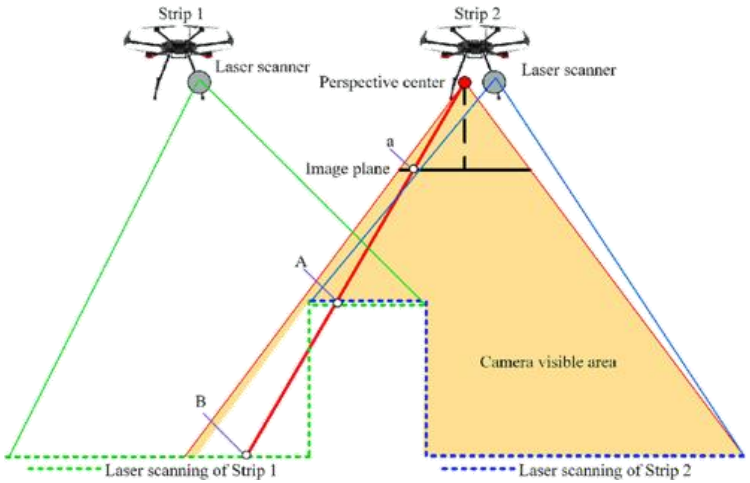
Loïca Avanthey Laurent Beaudoin

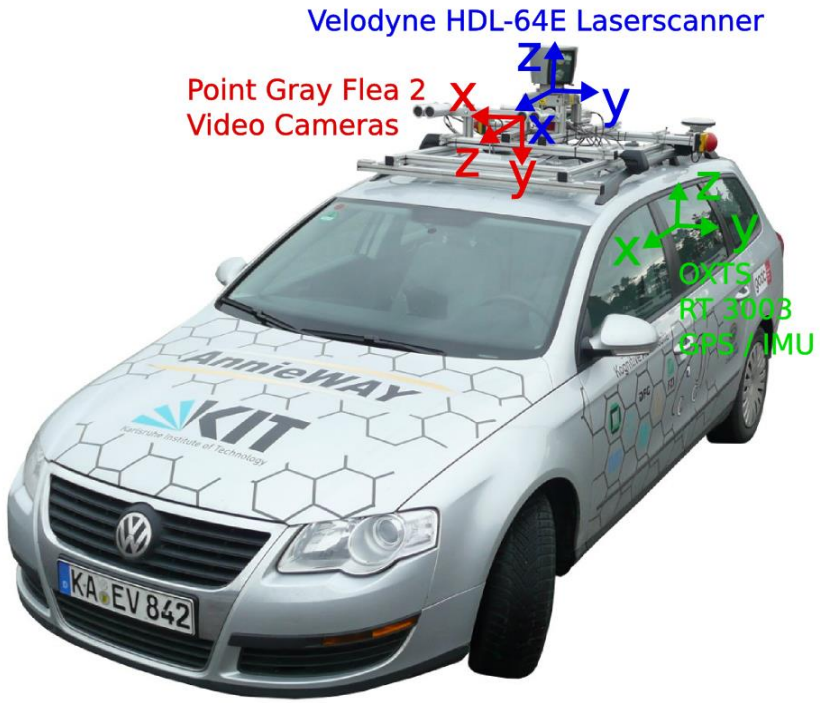
Nicolas Boutry Thibault Lejemble

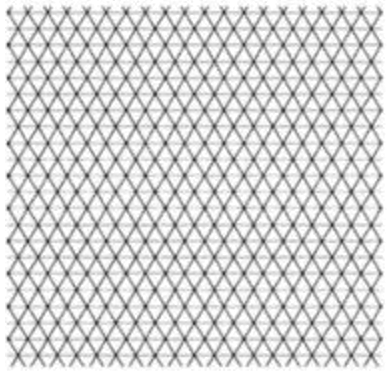


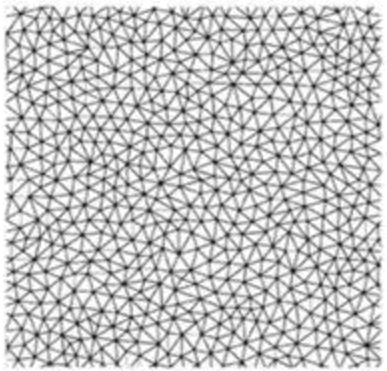
Contexte:





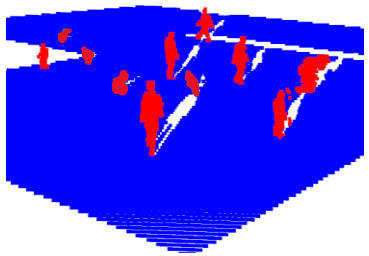






TESLA

State of the Art :

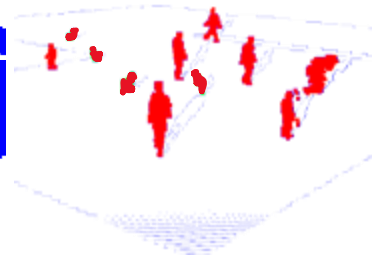


Captation:

Lidar

Camera Profondeur

Simulation



Filtering:

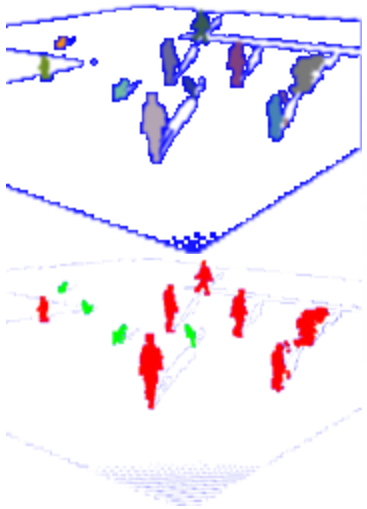
Statistique

KNN

Projection

Signal

Equation Différentielle

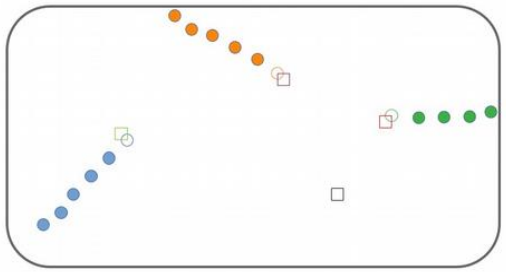


Segmentation:

PointNET++

Transformer

ASIS

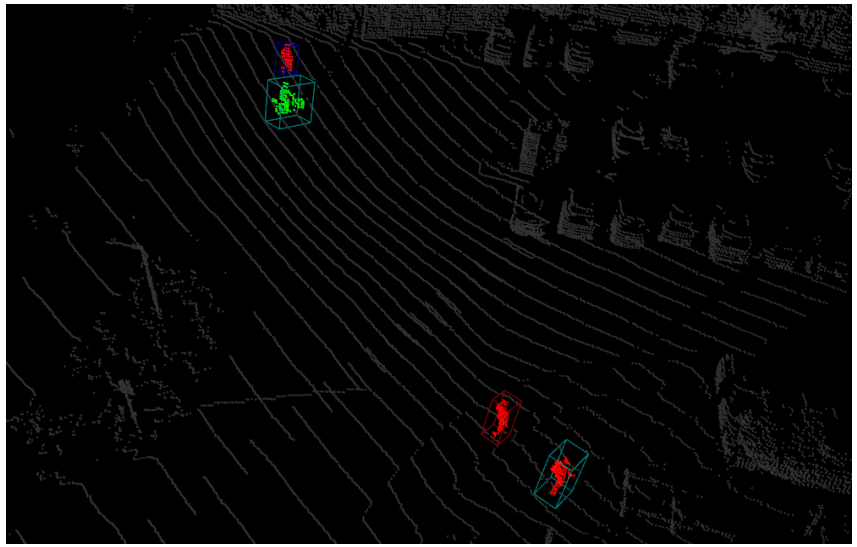


Tracking :

Kalmann Filter

MPBTrack

Prediction
Association



Résultat du PipeLine

A problematic : New and Realistic

Everything start with some data

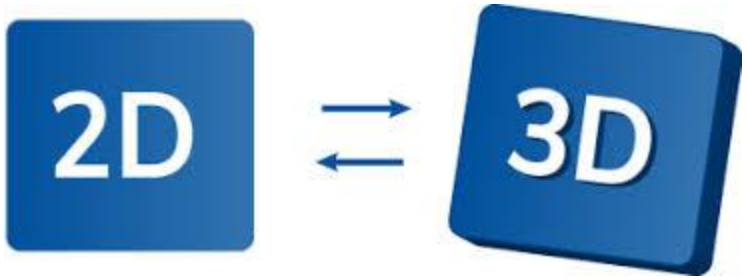
NL-Drive DHB

Kiti SemanticPoss

NuScene SuperCaustic

Waymo ShapeNet

Some Idea



Problematic and Protocol

Can automonous driven cars drive all around the world with a single model ?

Train on US datasets, test on English and even Indian datasets, and then exchange.

Study of the applicability of a situation for each trained model.

Why didn't it worked ?

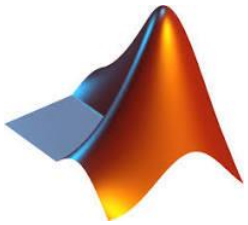
IDD-3D: Indian Driving Dataset in 3D Unstructured Road Scenes

Toolkit code, Data demos and Dataset coming soon!

2011_09_29_drive_0071 (4.1 GB)
Length: 1065 frames (01:46 minutes)
Image resolution: 1392 x 512 pixels
Labels: 0 Cars, 0 Vans, 0 Trucks, 0 Pedestrians, 0 Sitters, 0 Cyclists, 0 Trams, 0 Misc
Downloads: [unsynced+unrectified data] [synced+rectified data] [calibration]

conda create -n spconv2 python=3.9
conda activate spconv2
pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio==0.8.1 -f https://download.pytorch.org/whl/torch_stable.html
pip install numpy==1.19.5 protobuf==3.19.4 scikit-image==0.19.2 waymo-open-dataset-tf-2-5-0 nuscenes-devkit==1.0.5 spconv-cu111 numba scipy pyyaml easydict fire tqdm shapely matplotlib opencv-python addict pyquaternion awscli open3d pandas future pybind11 tensorboardX tensorboard Cython prefetch-generator

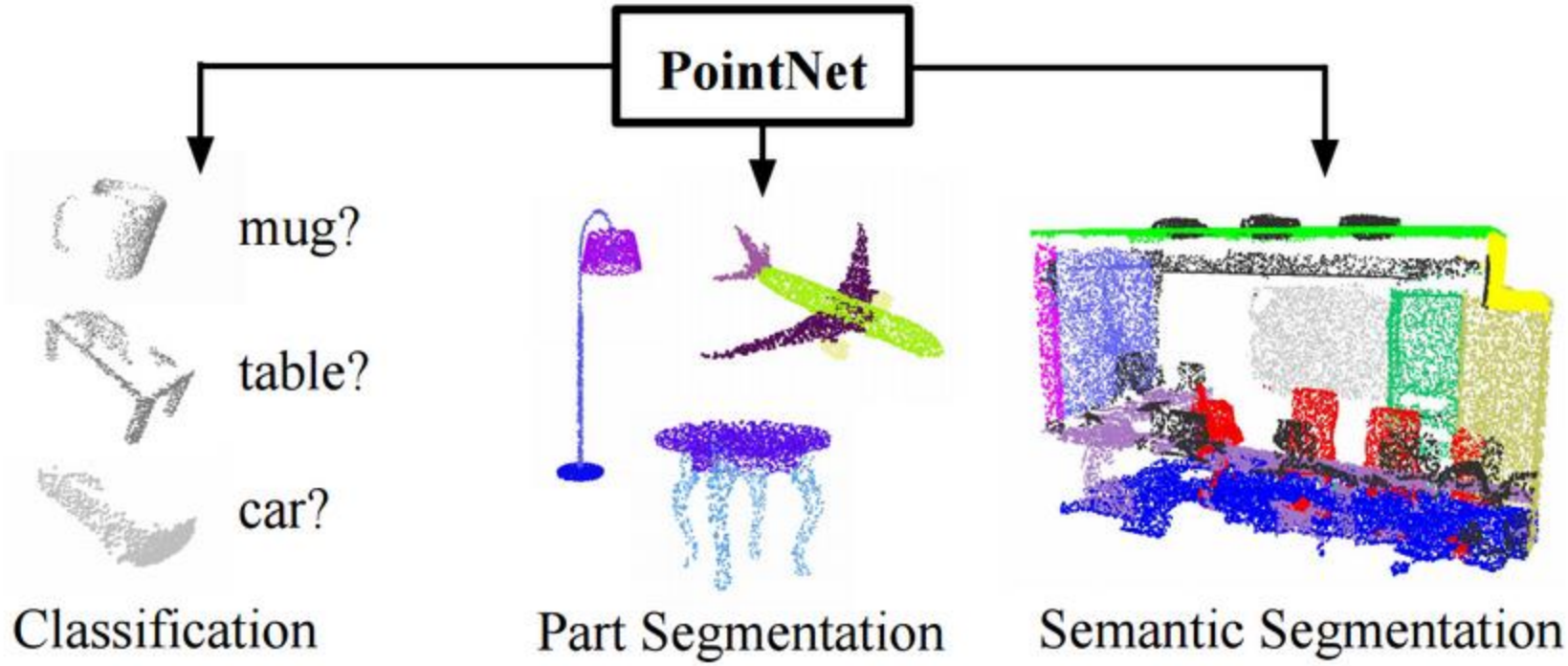
Google colab



PointNet From Scratch

Google colab

PyTorch



InnerWorking of PointNet

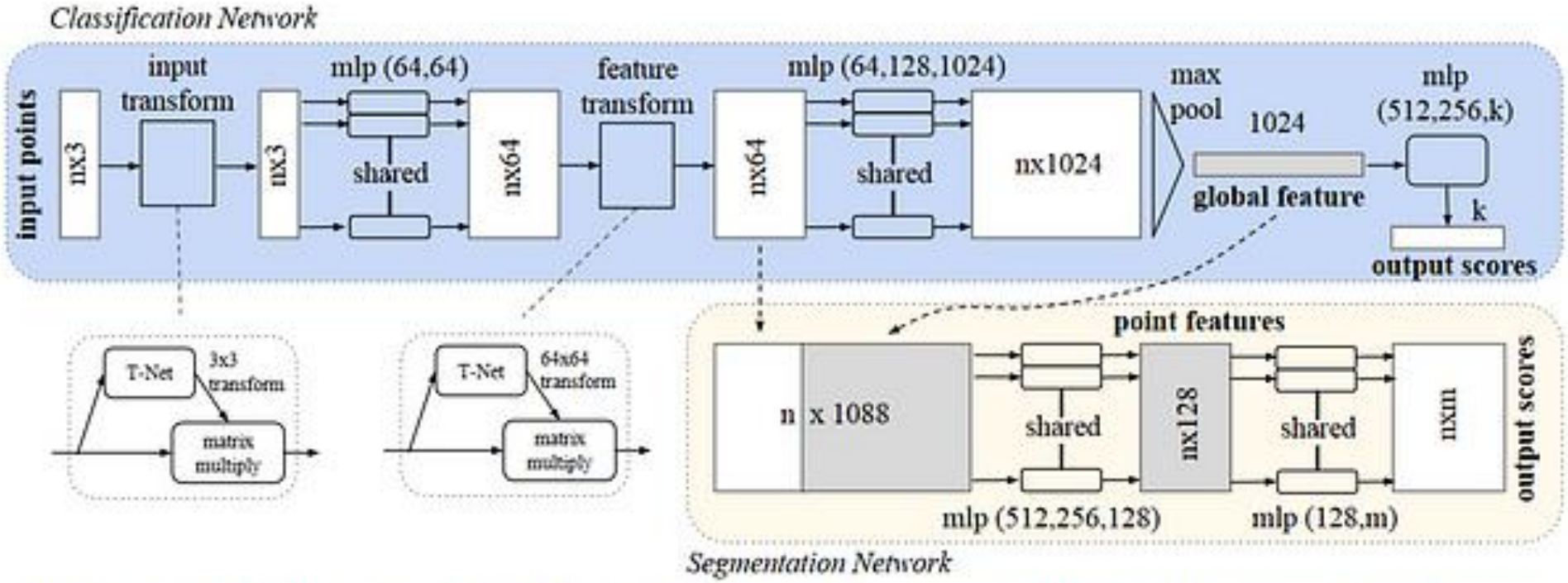
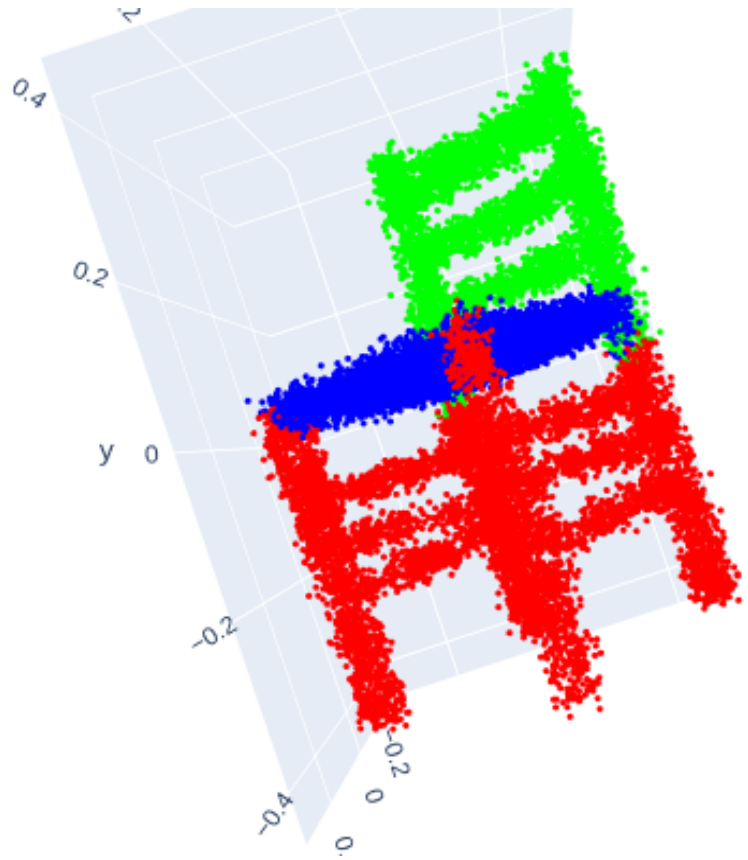


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

Results and Opening



References:

- Segmentation, Classification and Tracking of Objects in Lidar Point Cloud Data Using Deep Learninbg (ROBIN BERNSTA LE, HJALMAR LIND)
- A review of algorithms for filtering the 3D point cloud (Xian-Feng Han , et al.)
- Associatively Segmenting Instances and Semantics in Point Clouds (Xinlong Wang, et al.)
- PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation (Charles R. Qi, et al.)
- Point Transformer (Hengshuang Zhao, et al.)
- An Introduction to the Kalman Filter (Greg Welch, et al.)
- MBPTrack: Improving 3D Point Cloud Tracking with Memory Networks and Box Priors (Tian-Xing Xu, et al.)
- Is It Safe to Drive? An Overview of Factors, Metrics, and Datasets for Driveability Assessment in Autonomous Driving (Junyao Guo, et al.)
- Point Net from Scratch (Isaac Berrios)