Point cloud labeling using machine learning

Andrii Babii apratster@gmail.com
Data sources

1. LIDAR
2. 3D IR cameras
3. Generated from 2D sources

3D dataset: [http://kos.informatik.uni-osnabrueck.de/3Dscans/](http://kos.informatik.uni-osnabrueck.de/3Dscans/) by Dorit Borrmann from Jacobs University Bremen gGmbH, Germany.
Data sources

3D sensors not so expensive now.

Quality. Speed. Weight. Range. Angle

Velodine LeddarTech Riegl Routescene YellowScan Leica Geosystems Scanse

3D cameras and 3D Scanners
Point Cloud Datasets

http://www.semantic3d.net/
https://sites.google.com/site/kevinlai726/datasets
http://www.cvlibs.net/datasets/kitti/raw_data.php
https://sourceforge.net/projects/pointclouds/files/PCD%20datasets/

... 

and more
Pre-processing

Dense Voxel Grid:

A: Typical Voxel

B: Voxel Set

C: Voxel Grid

Read more and image from:

Huang J., You S. Point cloud labeling using 3d convolutional neural network
Pattern Recognition (ICPR), 2016 23rd International Conference on.— C. 2670-2675.
Pre-processing

**Problem** with Voxel Grid (Volumetric CNN):

1. Sparsity
2. Size (Memory usage raise up!)
3. Density approximation

**Solutions:**

1. FPNN
2. Spectral CNN
3. Multiview CNN
4. Spectral CNN
5. Feature-based DNN
6. Deep Learning on **Unordered Sets**
Labeling pipeline

But, return to Voxels for now:

Convolution ➔ Features maps ➔ Max-Pool ➔ Fully-connected layer
Why convolution?

https://hcordeirodotcom.files.wordpress.com/2012/02/atividade-2-kernel-convolution.jpg
Convolution in 3D

\[ v^{xyz}_{l,m} = b_{l,m} + \sum_{q} \sum_{i=0}^{f-1} \sum_{j=0}^{f-1} \sum_{k=0}^{f-1} w_{lmq}^{ijkl} q(x+i)(y+j)(z+k) \]

https://www.autodeskresearch.com/blog/kdd-2016-autodesk-research
Let’s remember - this is classical **CNN**, why not DNN?
3D convolution neural network


Let’s remember - this is classical CNN, why not DNN?

Sparsity! Memory size!
Pre-processing

What if we do not want to use Dense Voxel Grid?

There are 3 ways to deal with unordered sets:

1) Sort input into a canonical order;

2) Treat the input as a sequence to train an RNN, but augment the training data by all kinds of permutations;

3) Use a simple symmetric function to aggregate the information from each point. Symmetric function takes $n$ vectors as input and outputs a new vector that is invariant to the input order.
Readings:

3. Xu Xu and Sinisa Todorovic. Beam Search for Learning a Deep Convolutional Neural Network of 3D Shapes
4. A Lightweight 3D Convolutional Neural Network for Real-Time 3D Object Recognition
10. Martin Simonovsky, Nikos Komodakis Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs.