Plan for 3D Deep

Jungwon Kang, Maryam Jameela, Razieh Ramak

Sept 30 2018

Objectives

- Building 1st Version Deep Network for Each Task

Task	Major Contributor	Objectives
Noise filtering (for Optech)	Razieh Ramak	Point cloud segmentation (Noise/Non-noise), non real-time
Point cloud segmentation (for Optech)	Maryam Jameela	Point cloud segmentation (N-class objects), non real-time
3D object detection (for Thales)	Jungwon Kang	Real-time 3D object detection

Schedule

Month	Task	Deliverable
Oct 2018	• Problem definition • Dataset preparation • Literature survey	• Document describing problem definition, dataset, and literature survey • Visualization of dataset
Nov	• Practicing deep library • Design & implementation	• Document describing design
Dec	• Implementation	• Source code (Dec 31)
Jan 2019	• Documentation	• Document describing implementation (Jan 15)

*Submission deadline of major conferences starts from March.

Management Policy

- Regular meeting or discussion biweekly

- Team website:
  - https://github.com/yorku-ausml/deep3d

To-do List

- Problem definition, including
  - Cause of noise (Razieh)
  - Object classes (Maryam, Jungwon)
- Dataset description, including
  - Existing Optech airborne dataset (Razieh)
  - Dataset size
  - Current repository
  - Visualization
- Etc
  - Finding point cloud label tool (for making ground-truth)
  - Finding visualization tool

Key Literature

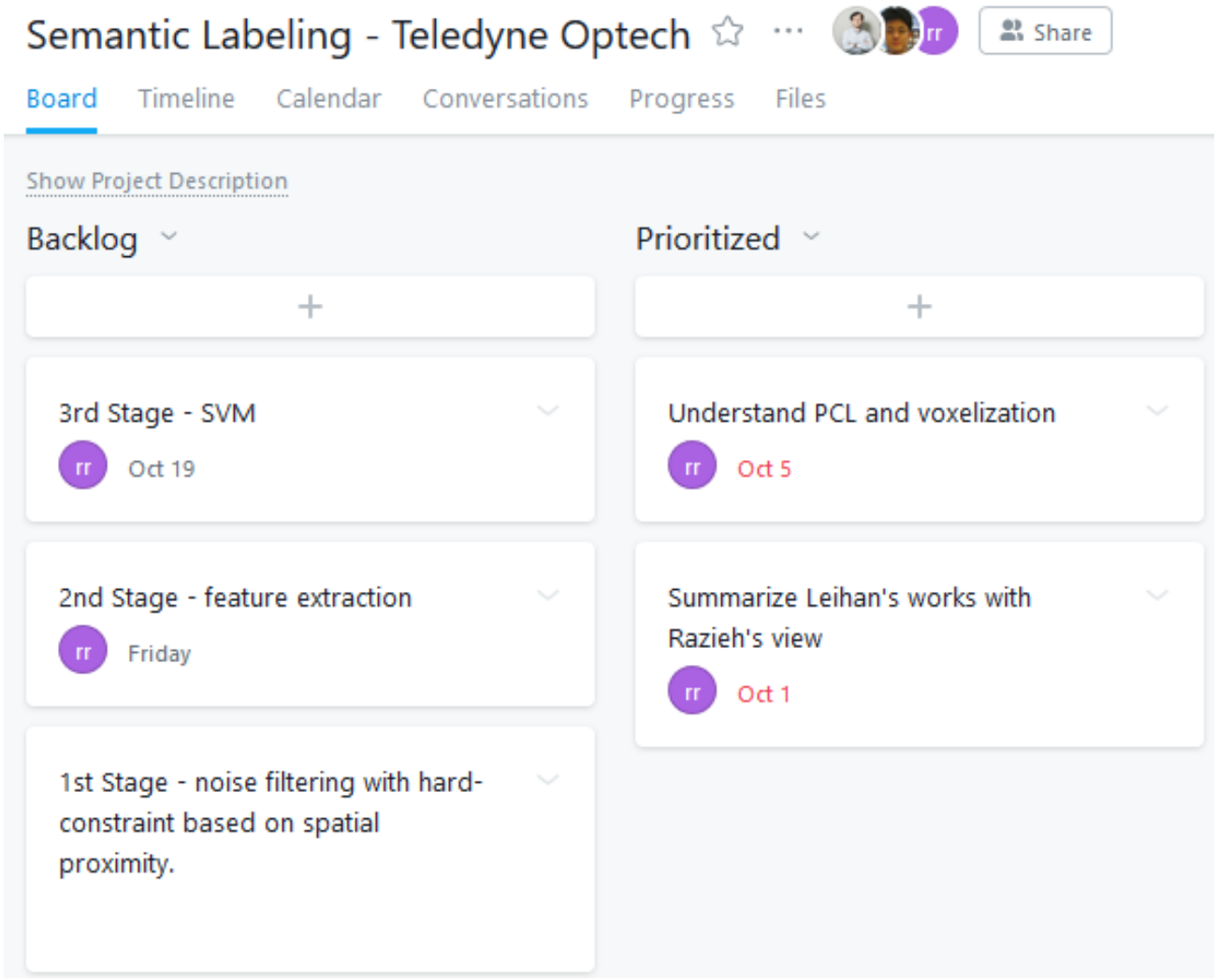
- Point cloud segmentation
  - Large-scale point cloud segmentation with superpoint graphs
    https://github.com/loicland/superpoint_graph
    *Rank 1 in http://www.semantic3d.net/
  - PointNet++: deep hierarchical feature learning on point sets in a metric space
    https://github.com/charlesq34/pointnet2
    *Rank 4 in http://www.semantic3d.net/
- Object detection
  - Joint 3D proposal generation and object detection from view aggregation
    https://github.com/kujason/avod

*Literature list is also available at https://github.com/yorku-ausml/deep3d/wiki/Related-works

Current Progress

Oct 12 2018

Asana Assignment



Progress on Noise Filtering

Razieh

- Atmospheric noise filtering
  - Noise filtering
    Segmentation of raw point cloud using voxelization
    Pre-classification by defining special rules
  - Feature extraction
    Using Eigen library and programming
  - Classification sing SVM
    Using "libSVM"

- Understanding PCL and voxelization
  - PCL
    A large scale, open project for 2D/3D image and point cloud processing. However, there is no PCL in noise filtering application
  - Voxelization
    A data structure used to represent a collection of multi-dimensional points and is commonly used to represent three-dimensional data

Clarifying the Task

Dataset used for training?
What kind of dataset will be used? Mobile data / airborne lasers / hybrid dataset mixture of both.
What kind of environment? Indoor / Outdoor or Urban / Rural / Forest
Which object classes?

Key Papers

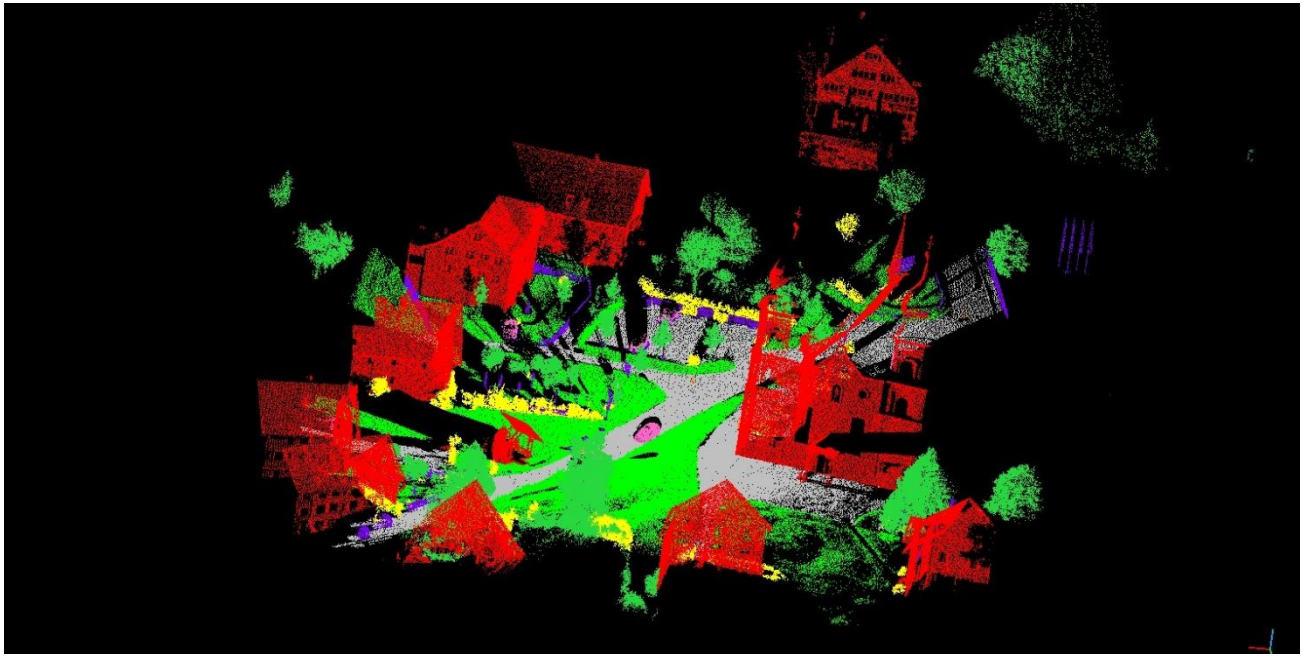
- Point cloud segmentation
  - PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation
    https://github.com/charlesq34/pointnet
    *Both used in the two following papers

  - PointNet++: deep hierarchical feature learning on point sets in a metric space
    https://github.com/charlesq34/pointnet2
    *Rank 4 in http://www.semantic3d.net/

  - Large-scale point cloud segmentation with superpoint graphs
    https://github.com/loicland/superpoint_graph
    *Rank 1 in http://www.semantic3d.net/

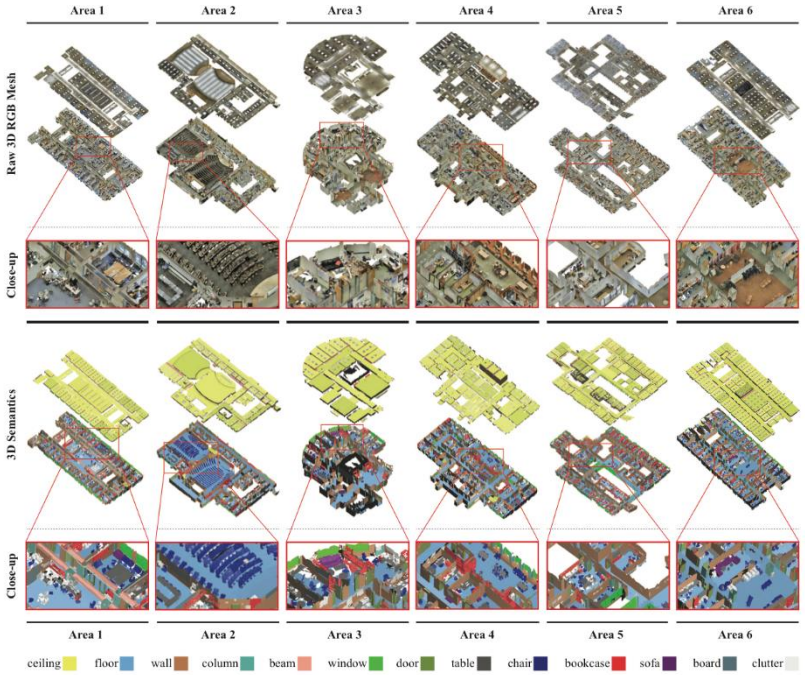
Publicly Available Dataset (1/3)

- Semantic3D
  - LiDAR dataset with over 3 billion points from a variety of urban and rural scenes.
  - http://www.semantic3d.net/
  - Managed by ETH (http://www.prs.igp.ethz.ch/)
  - 8 class labels, namely {1: man-made terrain, 2: natural terrain, 3: high vegetation, 4: low vegetation, 5: buildings, 6: hard scape, 7: scanning artefacts, 8: cars}.



Publicly Available Dataset (2/3)

- S3DIS (Stanford Large-Scale 3D Indoor Space)
  - 3D RGB point clouds of six floors from three different buildings
  - http://buildingparser.stanford.edu/dataset.html
  - Currently, *2D-3D-S dataset* is newly released.
  - 13 object classes (ceiling, floor, wall, beam, column, window, door, and movable elements: table, chair, sofa, bookcase, board and clutter for all other elements)

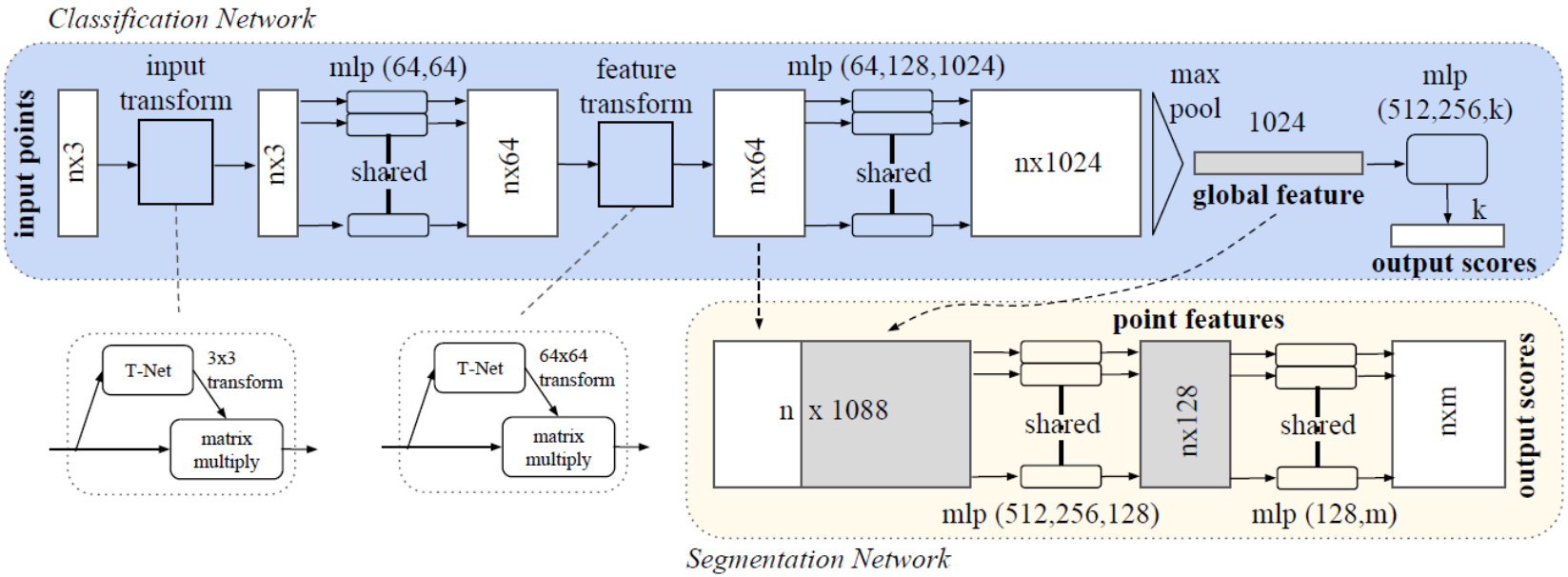


Publicly Available Dataset (3/3)

- Etc
  - Oakland 3-D Point Cloud Dataset (2009)
    - http://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/
  - NYU Depth Dataset V2 (2012)
    - https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
  - Sydney Urban Objects data set
    - http://www.acfr.usyd.edu.au/papers/SydneyUrbanObjectsDataset.shtml
  - IQmulus & TerraMobilita Contest
    - Mobile laser scans (MLS) in dense urban environments
    - http://data.ign.fr/benchmarks/UrbanAnalysis/
  - Vaihingen3D airborne benchmark
    - http://www2.isprs.org/commissions/comm3/wg4/3d-semantic-labeling.html

PointNet

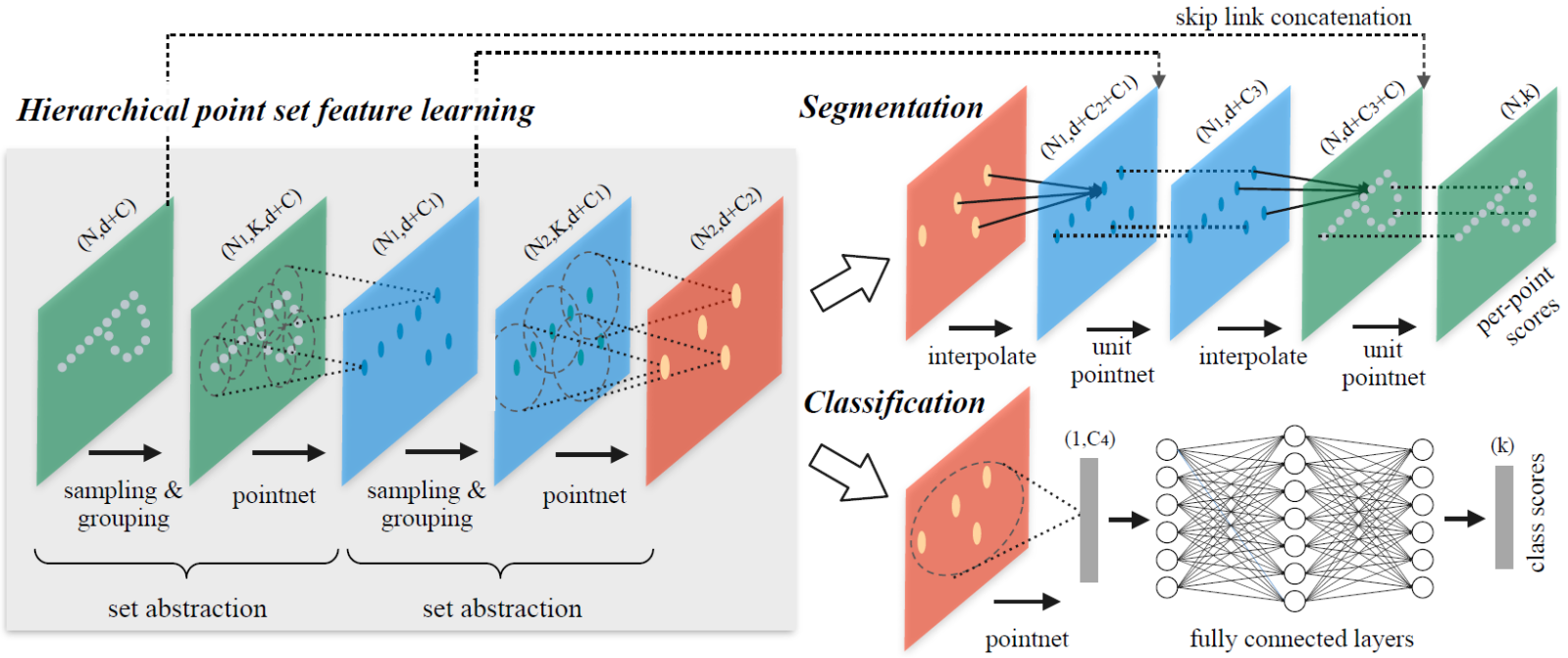
- Architecture
  - http://stanford.edu/~rqi/pointnet/



CVPR 2017

PointNet++

- Hierarchical Feature Learning Architecture
  - http://stanford.edu/~rqi/pointnet2/



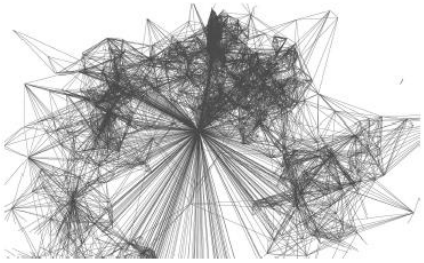
NIPS 2017

Superpoint Graph (1/2)

- Individual steps in pipeline

(a) RGB point cloud

(b) Geometric partition

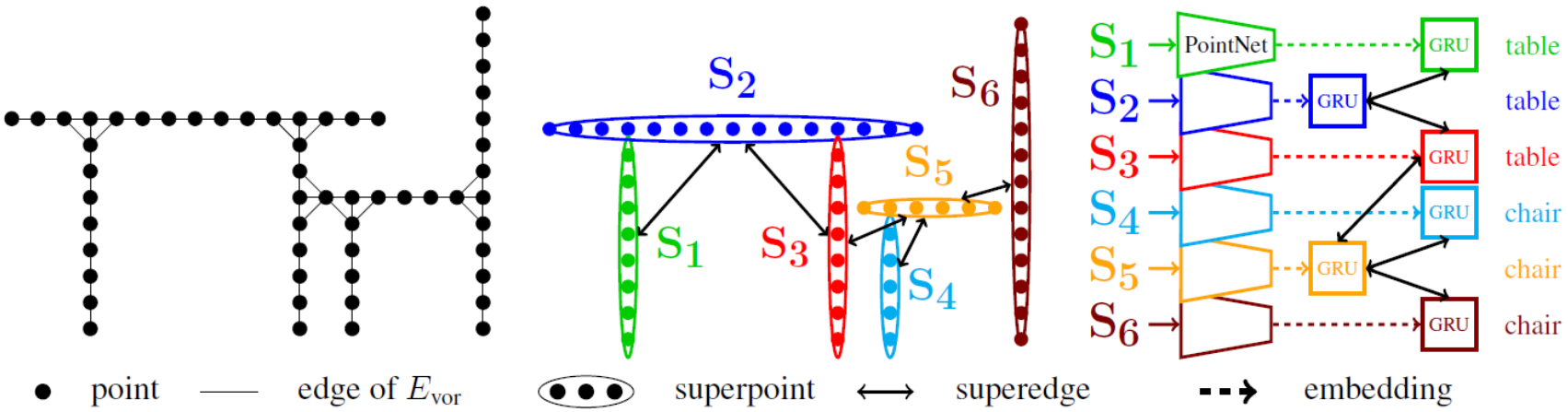


(c) Superpoint graph

(d) Semantic segmentation

CVPR 2018

Superpoint Graph (2/2)



(a) Input point cloud (b) Superpoint graph (c) Network architecture

Illustration of our framework on a toy scan of a table and a chair. We perform geometric partitioning on the point cloud (a), which allows us to build the superpoint graph (b). Each superpoint is embedded by a PointNet network. The embeddings are then refined in GRUs by message passing along superedges to produce the final labeling (c).

CVPR 2018

Plan

Month	Task	Deliverable
Oct 2018	• Problem definition • Dataset preparation • Literature survey	• Document describing problem definition, dataset, and literature survey • Visualization of dataset
Nov	• Practicing deep library • Design & implementation	• Document describing design
Dec	• Implementation	• Source code (Dec 31)
Jan 2019	• Documentation	• Document describing implementation (Jan 15)