3D Sensor Data Processing Curriculum

No.	Title	Content	Date	Speaker
01	3D Point Cloud	• 3D Point Clouds • Point Cloud Processing	08.13	이용이 (SOSLAB)
02	3D Data Acquisition (Passive)	• Stereo Vision • Photogrammetry & Multiview Geometry	08.20	박준휘 (MagicLeap)
03	3D Data Acquisition (Active)	• RGB-D Camera • LiDAR • Projector-Camera System	08.27	함승록 (LG전자)
04	Differential Geometry	• Differential Geometry	09.03	장승호 (MORAI)
05	Spatial Transformation	• Spatial Transformation	09.03	이용이 (SOSLAB)
06	Point Cloud Analysis #1	• Filtering • Nearest Neighbor Search	09.10	길현재 (서울대학교)
07	Point Cloud Analysis #2	• Model Fitting • Point Cloud Features	09.10	윤형석 (CMES)
08	Point Cloud Analysis #3	• Clustering	09.17	신동훈 (SOSLAB)
09	Point Cloud Analysis #4	• Classification and Segmentation • Registration	09.24	최재우 (PLAIF)
10	Point Cloud Analysis #5	• Deep Learning on Point-cloud	09.24	이종록 (Vueron Technology)
11	Point Cloud Analysis #6	• Communication • Visualization	10.08	이상운 (Seoul Robotics)
12	PCD Tools	• PCL • Open3D • CloudCompare	10.08	최준호 (SOSLAB)

Overview of Deep Learning on Point-cloud

2023.09.24

이종록 (lrrghdrh@naver.com)

Contents

Introduction

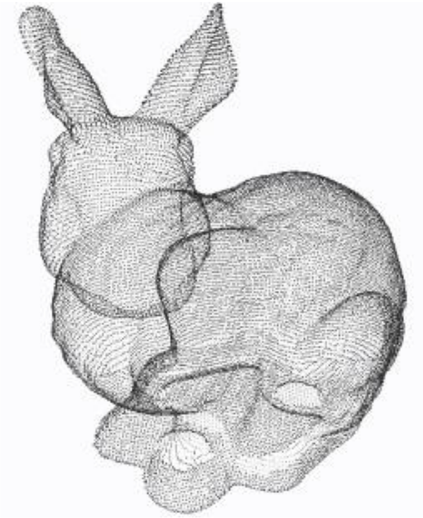
Image (2D) vs Point Cloud (3D)

- Image (RGB)
  - 데이터 타입: uint8
  - 데이터 형태: 배열 (index에 따른 위치가 보장)
  - 차원: HxWx3

1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

- Point Cloud
  - 데이터 타입: float32
  - 데이터 형태: 배열 집합 (순서가 중요하지 않음)
  - 차원: Nx4 (x, y, z, i)

Unstructured, Irregular

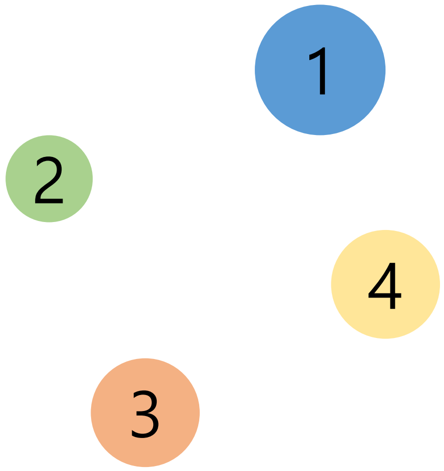


https://www.youtube.com/watch?v=gm_oW0bdzHs

Introduction

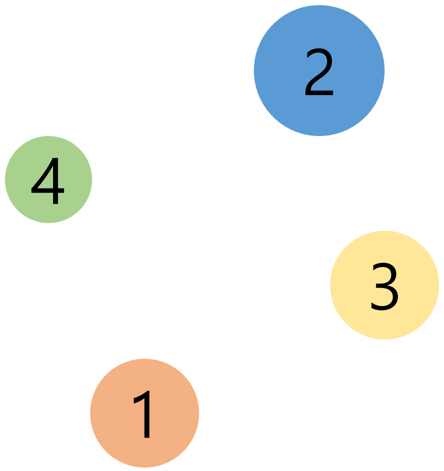
Image (2D) vs Point Cloud (3D)

Point Cloud 1

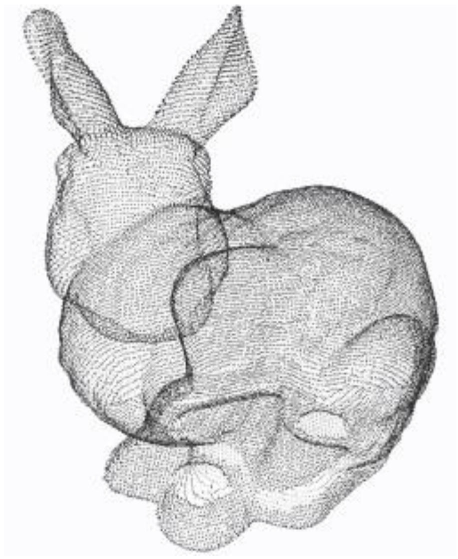


$$\begin{matrix}[[x_b, y_b, z_b] \\ [x_g, y_g, z_g] \\ [x_o, y_o, z_o] \\ [x_y, y_y, z_y]]\end{matrix}$$

Point Cloud 2



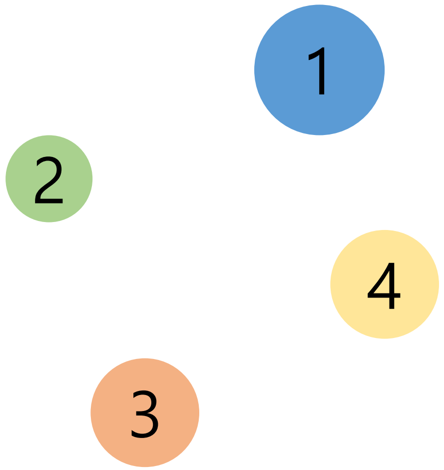
$$\begin{matrix}[[x_o, y_o, z_o] \\ [x_b, y_b, z_b] \\ [x_y, y_y, z_y] \\ [x_g, y_g, z_g]]\end{matrix}$$



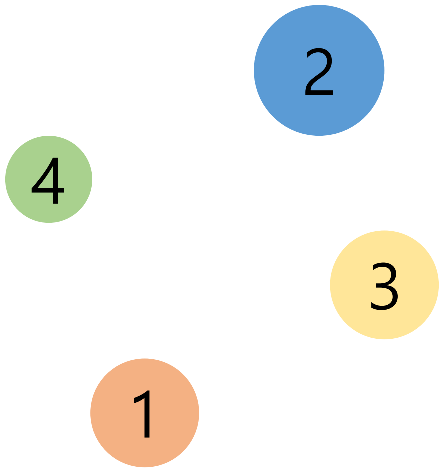
Introduction

Image (2D) vs Point Cloud (3D)

Point Cloud 1



Point Cloud 2



K1	K2	K3	K4

Conv1D(4)

$$
\begin{aligned}
\mathrm{F1} = \mathrm{K1} &\times [x_b, y_b, z_b] \\
+ \mathrm{K2} &\times [x_g, y_g, z_g] \\
+ \mathrm{K3} &\times [x_o, y_o, z_o] \\
+ \mathrm{K4} &\times [x_y, y_y, z_y]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{F2} = \mathrm{K1} &\times [x_o, y_o, z_o] \\
+ \mathrm{K2} &\times [x_b, y_b, z_b] \\
+ \mathrm{K3} &\times [x_y, y_y, z_y] \\
+ \mathrm{K4} &\times [x_g, y_g, z_g]
\end{aligned}
$$

$\mathrm{F1} \neq \mathrm{F2}$

Introduction

Image (2D) vs Point Cloud (3D)

- 즉, 기존의 방식으로는 Point Cloud 자체를 딥러닝의 Input으로 사용하기 어려움
  - Permutation variance: 입력 Index의 순서에 따라 결과가 달라짐
  - Point Cloud는 집합 (Set) 데이터이기 때문

F1 = K1 × $[x_b, y_b, z_b]$
+ K2 × $[x_g, y_g, z_g]$
+ K3 × $[x_o, y_o, z_o]$
+ K4 × $[x_y, y_y, z_y]$

F2 = K1 × $[x_o, y_o, z_o]$
+ K2 × $[x_b, y_b, z_b]$
+ K3 × $[x_y, y_y, z_y]$
+ K4 × $[x_g, y_g, z_g]$

F1 $\neq$ F2

Point Cloud Representation

Point Cloud Representation

- Point Cloud를 딥러닝에 사용하기 위한, 일종의 Pre-processing 혹은 표현 method
    - Voxel-based
    - Image-based
    - Point-based
    - Graph-based

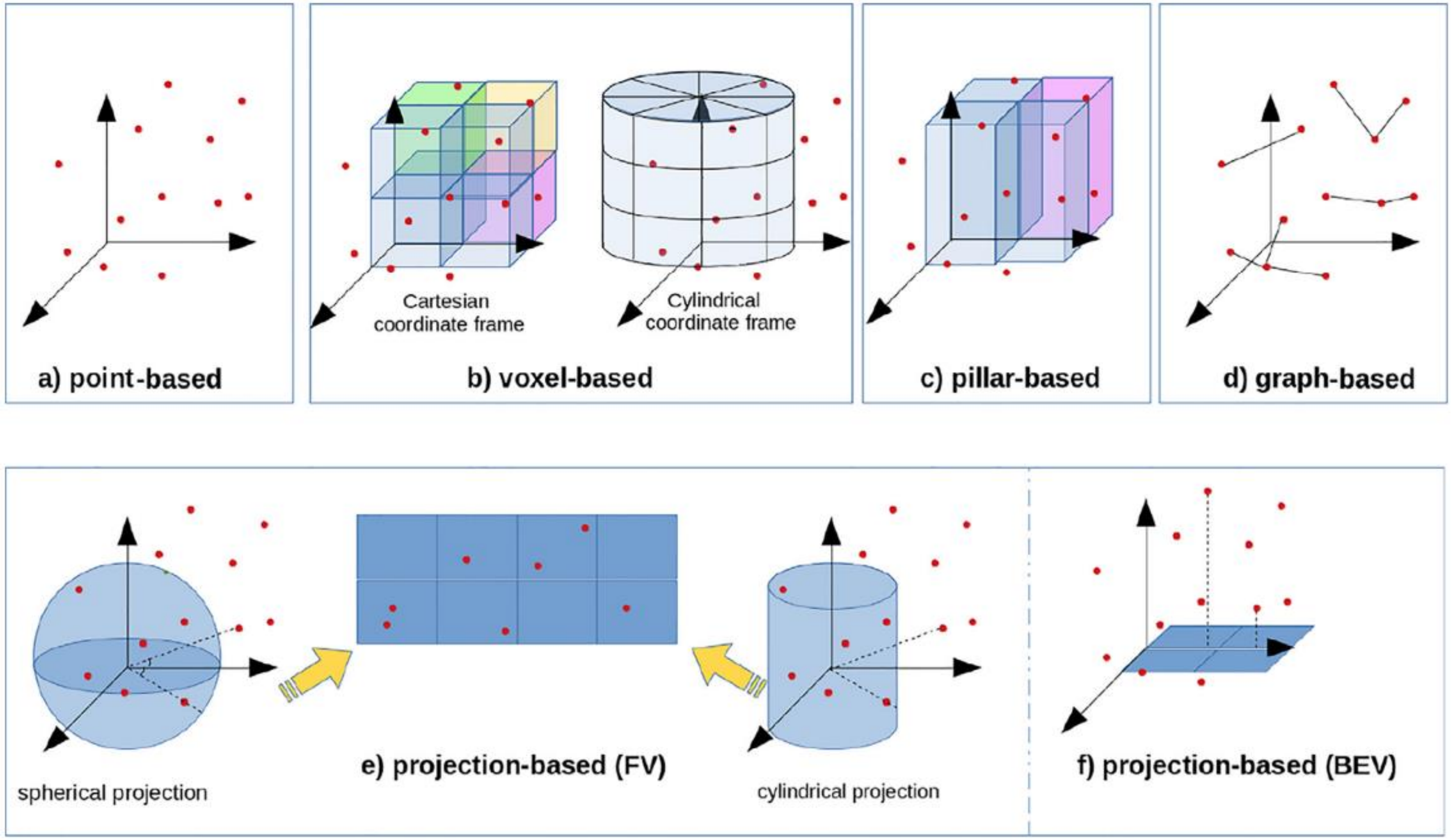


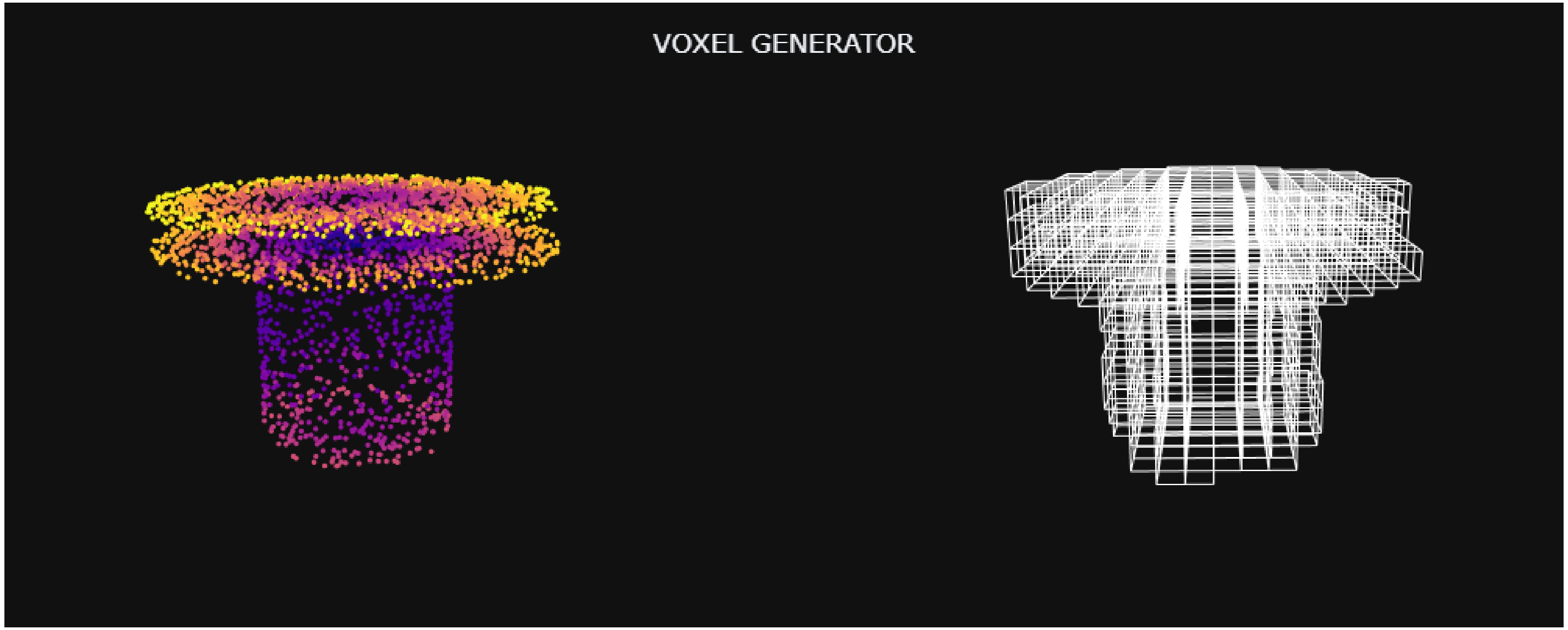
Fig. 2. Point cloud representations.

"A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving"

Point Cloud Representation

Voxel-based

- 3D Point Cloud를 Voxel화 하여 Tensor로 구성 (Voxelization)
  - 2D Image와 같은 Tensor 형태로 CNN 적용이 가능 (2D CNN, 3D CNN)
  - Grid size로 Voxel 크기 조정 가능
  - Voxel 내부 Point의 Feature를 가공해 Voxel Feature로 사용

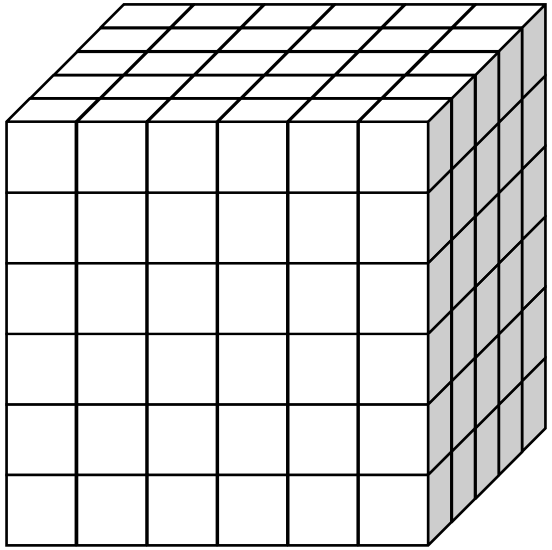


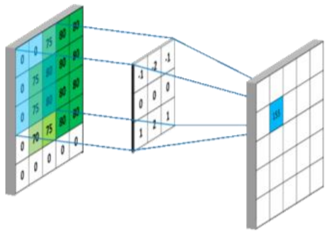
Point Cloud Representation

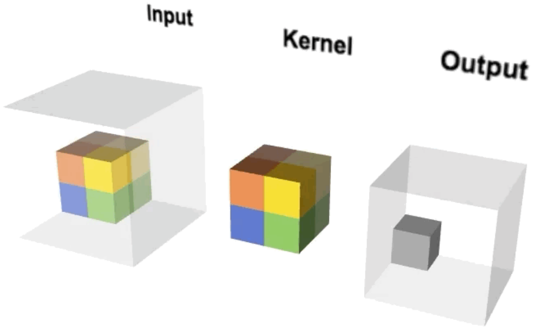
Voxel-based

- 장점:
  - 기존 Computer Vision 네트워크 (CNN) 적용이 용이

- 단점:
  - Voxelization으로 인한 Quantization Error 발생 (Continuous → Discrete)
  - 3D CNN 사용 시 많은 연산량 필요
    - 요즘은 SpConv의 등장으로 단점 보완





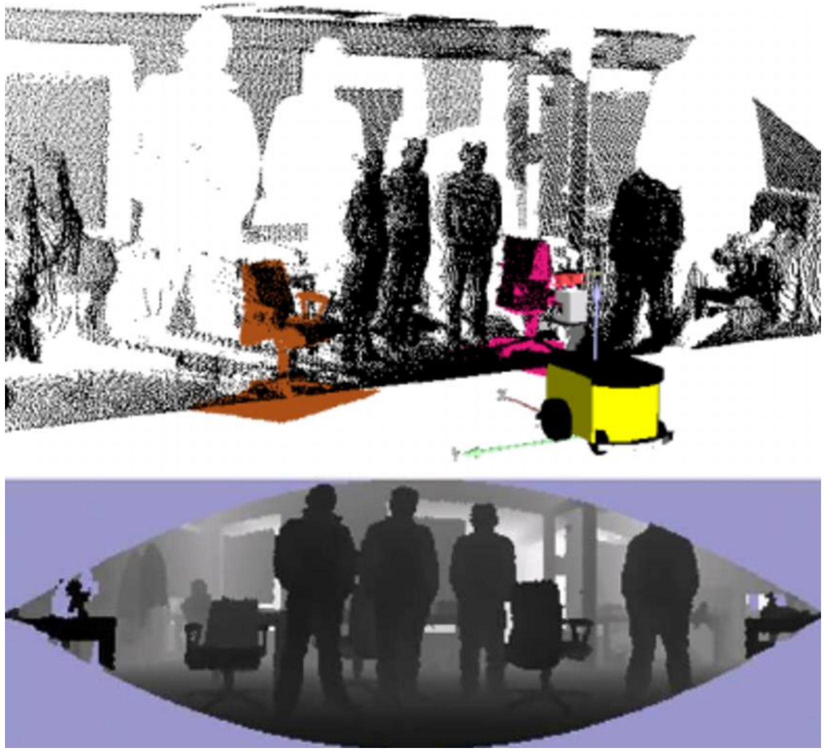


https://thomelane.github.io/convolutions/

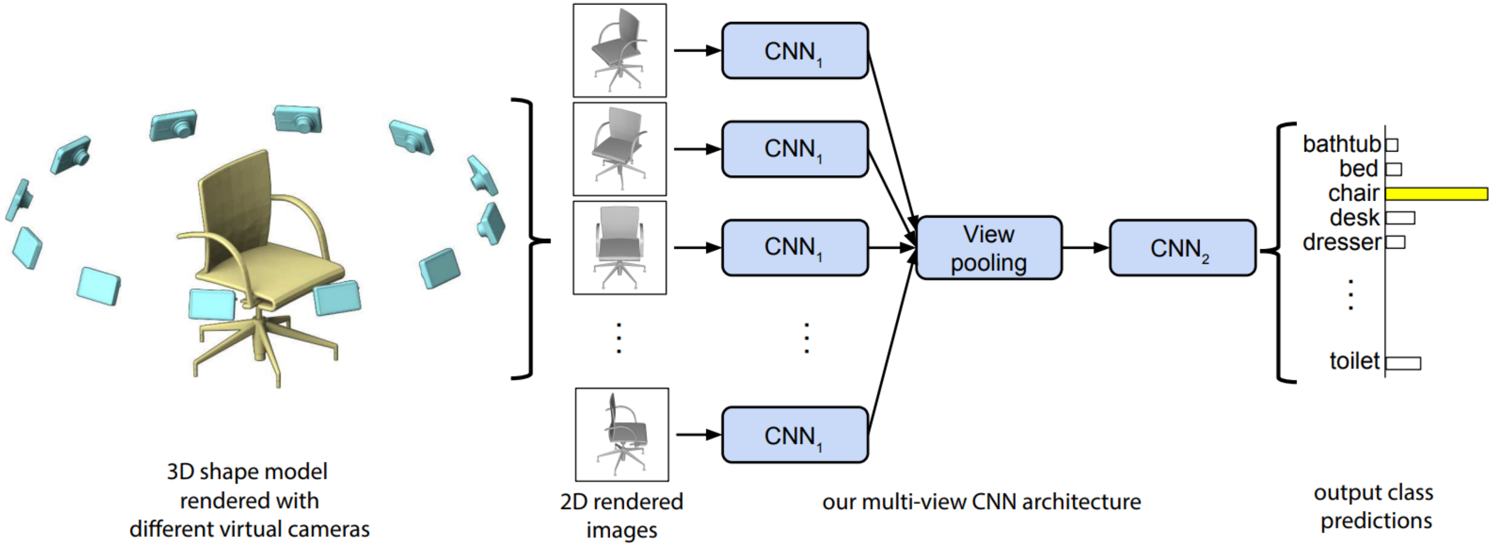
Point Cloud Representation

Image-based

- 3D Point Cloud를 다양한 view point에서의 Image 또는 Range Image로 변환
- Projection-based 라고도 불리움
- Image 형태로 CNN 적용 가능



"Robust On-line Model-based Object Detection"



"Multi-view convolutional neural networks for 3d shape recognition"

Point Cloud Representation

Image-based

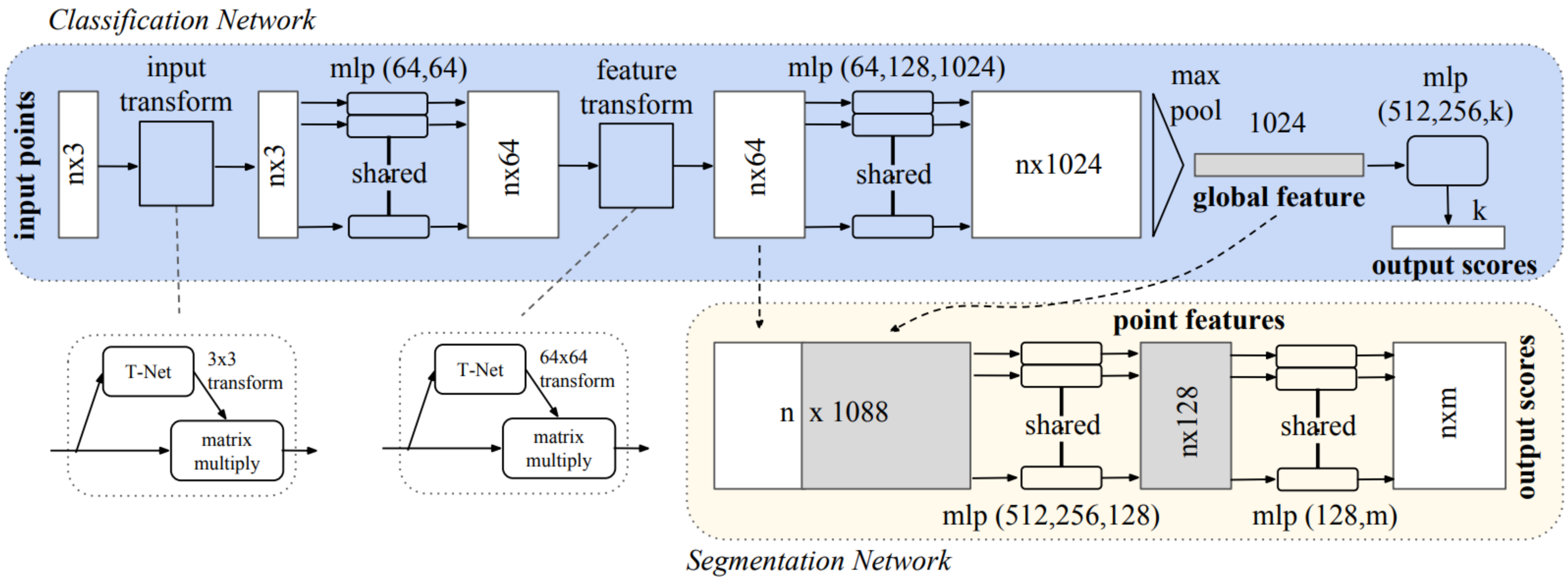
- 장점:
  - Voxel-based와 같이 Computer Vision 네트워크 (CNN) 적용이 용이
  - 2D CNN 사용으로 비교적 적은 메모리 사용, 빠른 네트워크 속도

- 단점:
  - Projection으로 인한 Quantization Error 발생 (Continuous → Discrete)
  - 데이터 표현력 한계 존재 (2D Image)

Point Cloud Representation

Point-based

- 3D Point Cloud 자체를 입력으로 사용하는 방식 (PointNet)
  - Permutation Invariance를 해결
  - Geometry Transform Invariance를 해결

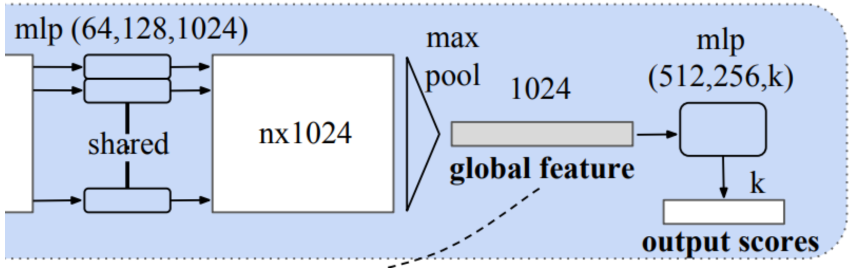


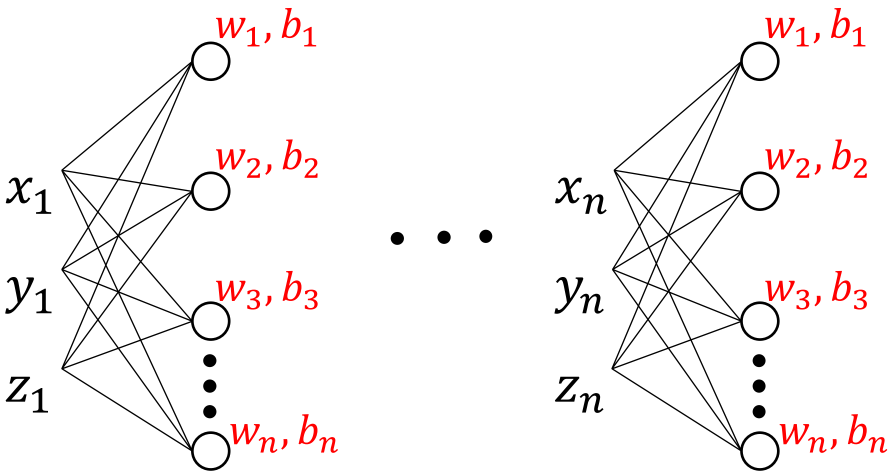
"PointNet: Deep learning on point sets for 3d classification and segmentation"

Point Cloud Representation

PointNet

- Permutation Invariance
  - Multi Layer Perceptron와 Max Pooling으로 해결
- Geometry Transform Invariance
  - Spatial Transform Network (STN)으로 해결





$N \times 1024$:

$$p_1 = [f_{1,1}, f_{2,1}, \dots, f_{1024,1}]$$
$$p_2 = [f_{1,2}, f_{2,2}, \dots, f_{1024,2}]$$
$$\dots$$
$$p_N = [f_{1,N}, f_{2,N}, \dots, f_{1024,N}]$$

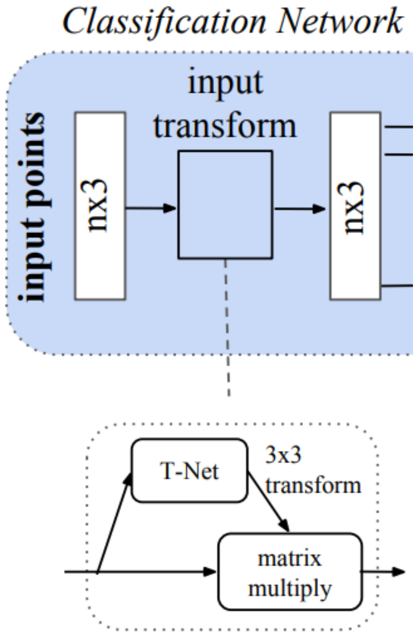
1024:

$$V = [\max(f_1), \max(f_2), \dots, \max(f_{1024})]$$

Point Cloud Representation

PointNet

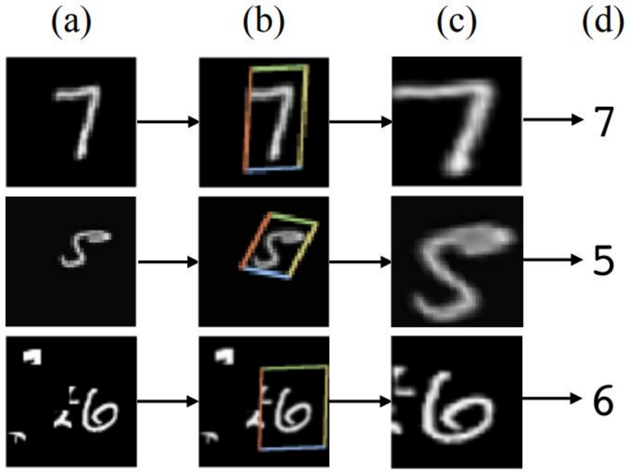
- Permutation Invariance
    - MLP와 Max Pooling으로 해결
- Geometry Transform Invariance
    - Spatial Transform Network (STN)으로 해결
    - 3D Space를 Transform에 Invariant한 Canonical Space로 변환



$$\mathrm{p} = [x, y, z]$$

$$T = \begin{pmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{pmatrix} \longrightarrow p_c = T \times \mathrm{p} = [x_c, y_c, z_c]$$

Canonical Space



"Spatial Transformer Networks"

Point Cloud Representation

PointNet

- 제안한 방법에 대한 근거 제시
  - 정리 1: Max Pooling과 MLP로 Point Cloud 특징을 표현할 수 있다. (Set 데이터에 적용 가능)
  - 정리 2: PointNet은 도형을 요약하는 방식으로 특징을 학습한다. (Noise나 Corruption에 Robust)

Theorem 1 *Suppose* $f : \mathcal{X} \rightarrow \mathbb{R}$ *is a continuous set function w.r.t Hausdorff distance* $d_H(\cdot, \cdot)$. $\forall \epsilon > 0$, $\exists$ *a continuous function* h *and a symmetric function* $g(x_1, \ldots, x_n) = \gamma \circ MAX$, *such that for any* $S \in \mathcal{X}$,

$$\left| f(S) - \gamma \left(\underset{x_i \in S}{MAX} \{h(x_i)\} \right) \right| < \epsilon$$

where $x_1, \ldots, x_n$ *is the full list of elements in* S *ordered arbitrarily,* γ *is a continuous function, and MAX is a vector max operator that takes* n *vectors as input and returns a new vector of the element-wise maximum.*

Theorem 2 *Suppose* $\mathbf{u} : \mathcal{X} \rightarrow \mathbb{R}^K$ *such that* $\mathbf{u} = \underset{x_i \in S}{MAX}\{h(x_i)\}$ *and* $f = \gamma \circ \mathbf{u}$. *Then,*

(a) $\forall S, \exists\, \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}$, $f(T) = f(S)$ *if* $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$;

(b) $|\mathcal{C}_S| \leq K$

"PointNet: Deep learning on point sets for 3d classification and segmentation"

Point Cloud Representation

PointNet

- 제안한 방법에 대한 수학적 근거 제시
  - 정리 1: Max Pooling과 MLP로 Point Cloud 특징을 표현할 수 있다. (Set 데이터에 적용 가능)
  - 정리 2: PointNet은 도형을 요약하는 방식으로 특징을 학습한다. (Noise나 Corruption에 Robust)

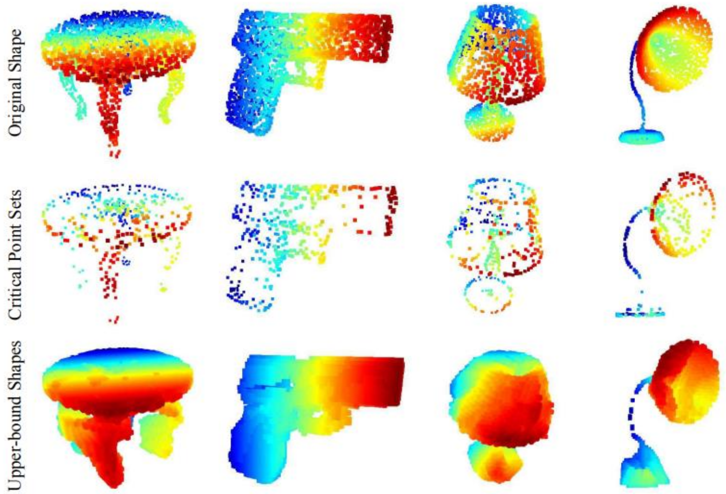


Figure 7. **Critical points and upper bound shape.** While critical points jointly determine the global shape feature for a given shape, any point cloud that falls between the critical points set and the upper bound shape gives exactly the same feature. We color-code all figures to show the depth information.

Theorem 2 *Suppose* $\mathbf{u} : \mathcal{X} \to \mathbb{R}^K$ *such that* $\mathbf{u} = \underset{x_i \in S}{MAX}\{h(x_i)\}$ *and* $f = \gamma \circ \mathbf{u}$*. Then,*

(a) $\forall S, \exists\, \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X},\ f(T) = f(S)$ *if* $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$;

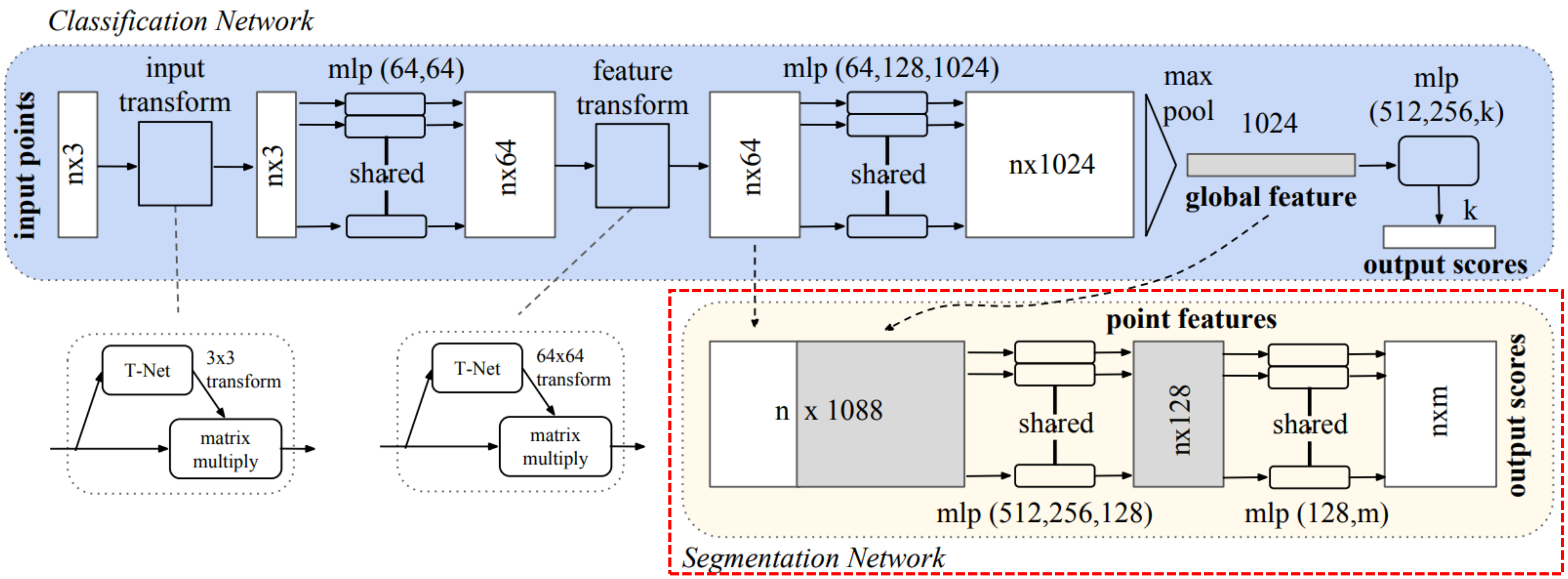
(b) $|\mathcal{C}_S| \leq K$

“PointNet: Deep learning on point sets for 3d classification and segmentation”

Point Cloud Representation

PointNet++

- PointNet의 단점을 극복
  - Global Feature는 학습하지만, Local Feature는 따로 학습하지 않는다.
  - Local Structure의 부재

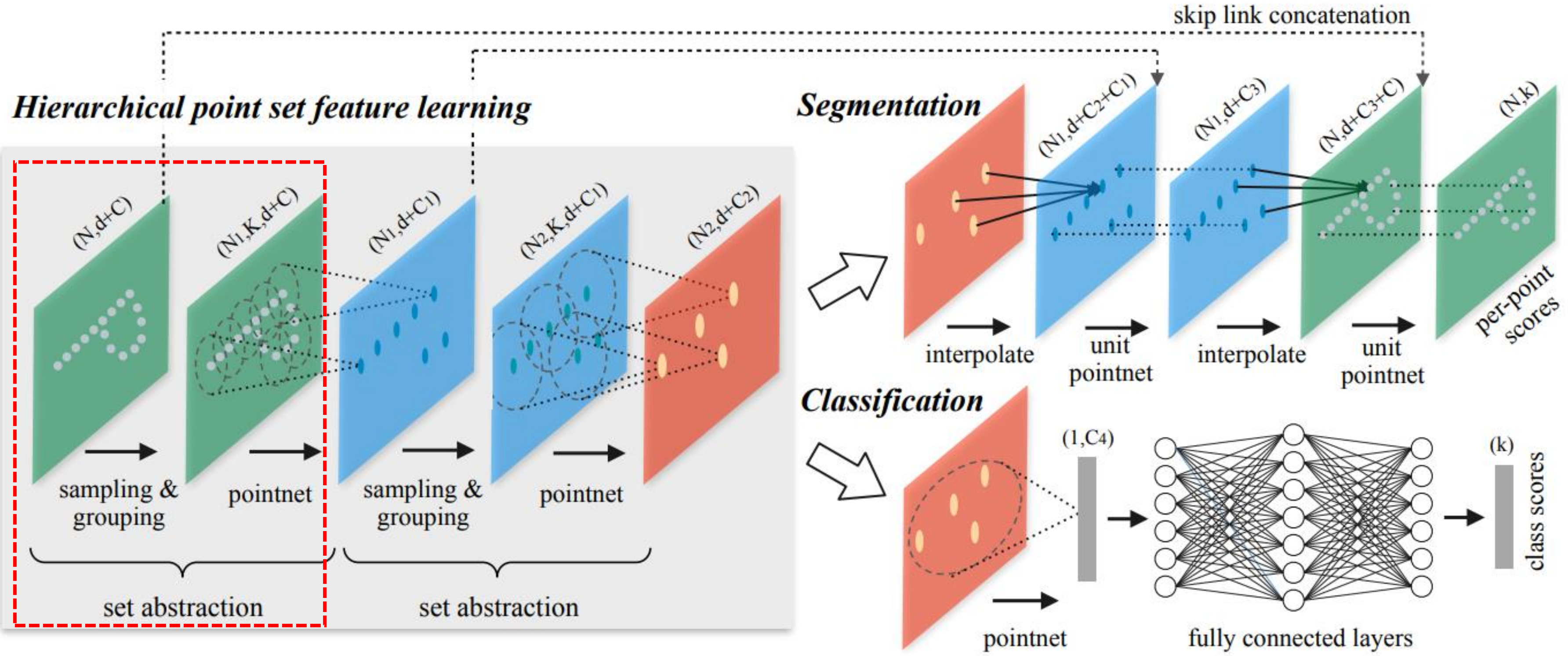


“PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”

Point Cloud Representation

PointNet++

- PointNet의 단점을 극복
  - Locally 하게 PointNet을 사용해, Local Structure를 구성 (Set Abstraction)

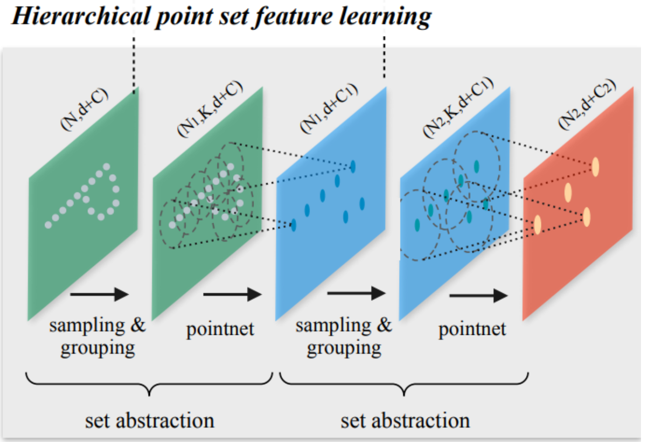


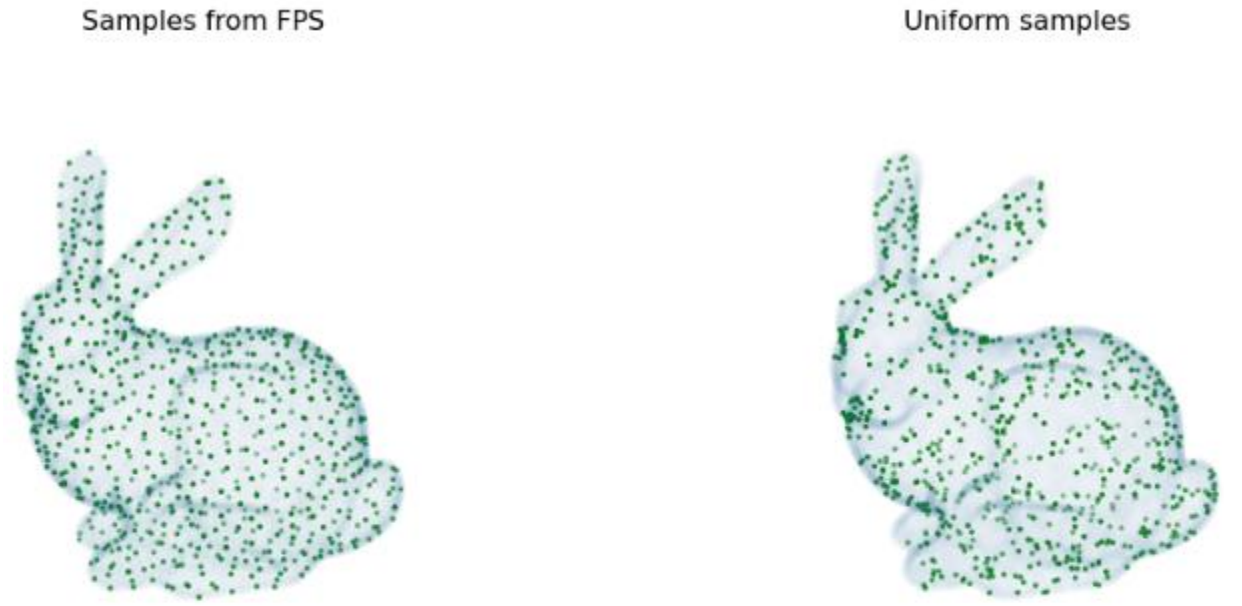
"PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space"

Point Cloud Representation

PointNet++

- Set Abstraction
  - Sampling: Farthest Point Sampling (FPS) [$N \times C \rightarrow N' \times C$]
  - Grouping: Ball Query (L2 Distance)
  - PointNet: MLP + Max Pooling



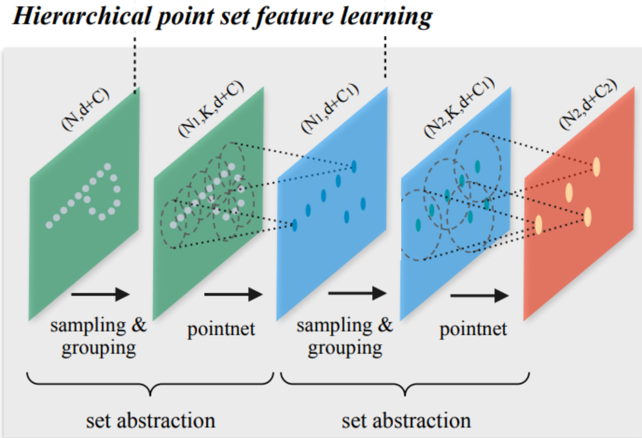


https://minibatchai.com/sampling/2021/08/07/FPS.html

Point Cloud Representation

PointNet++

- Set Abstraction
  - Sampling: Farthest Point Sampling (FPS) [N×C → N’×C]
  - Grouping: Ball Query (L2 Distance) [N’×C → N’×K×C]
  - PointNet: MLP + Max Pooling

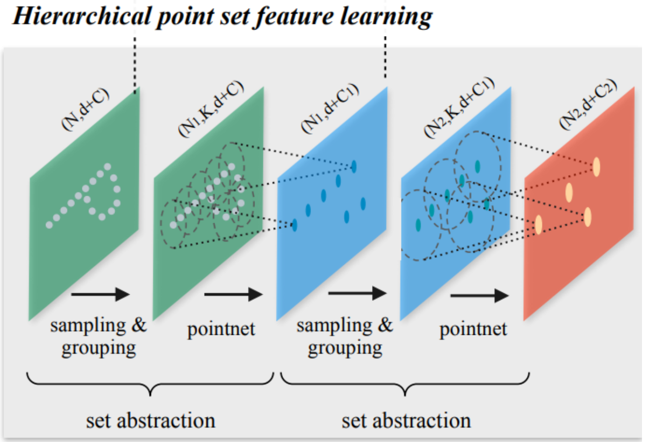


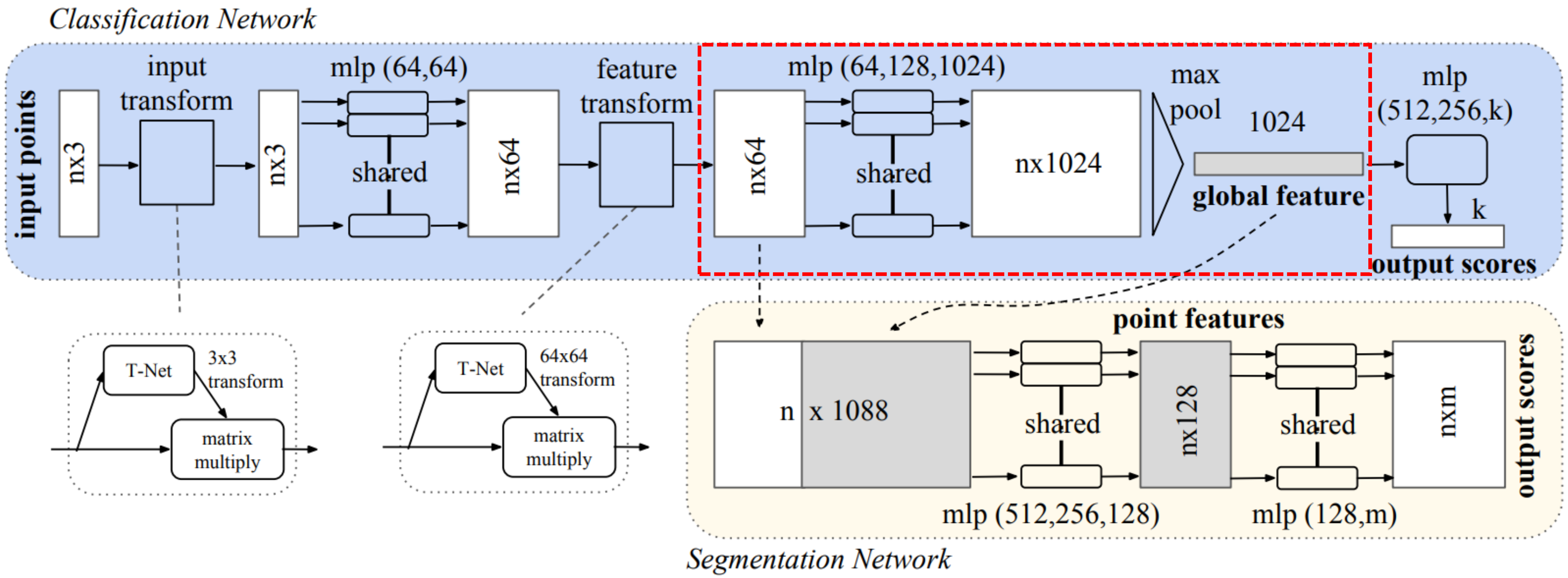
“Grid-GCN for Fast and Scalable Point Cloud Learning”

Point Cloud Representation

PointNet++

- Set Abstraction
  - Sampling: Farthest Point Sampling (FPS) [N×C → N’×C]
  - Grouping: Ball Query (L2 Distance) [N’×C → N’×K×C]
  - PointNet: MLP + Max Pooling [N’×K×C → N’×C’]



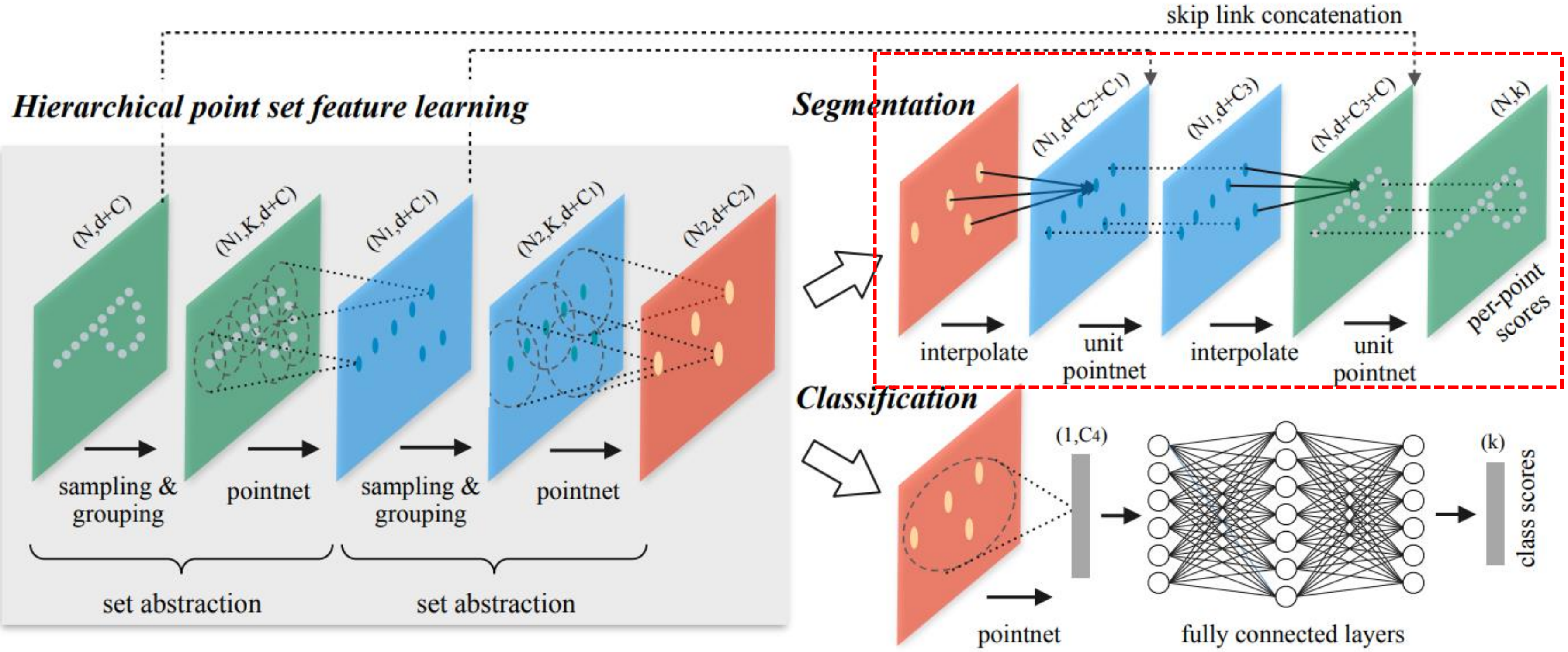


“PointNet: Deep learning on point sets for 3d classification and segmentation”

Point Cloud Representation

PointNet++

- Feature Propagation
  - SA에서 다운샘플링된 Point 개수를 Segmentation을 위해 원래의 N개로 복원
  - Skip Connection과 Inverse Distance Weighting (IDW) Interpolation으로 복원



"PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space"

Point Cloud Representation

Point-based

- 장점:
  - Point Cloud를 가장 직관적인 형태로 딥러닝에 사용

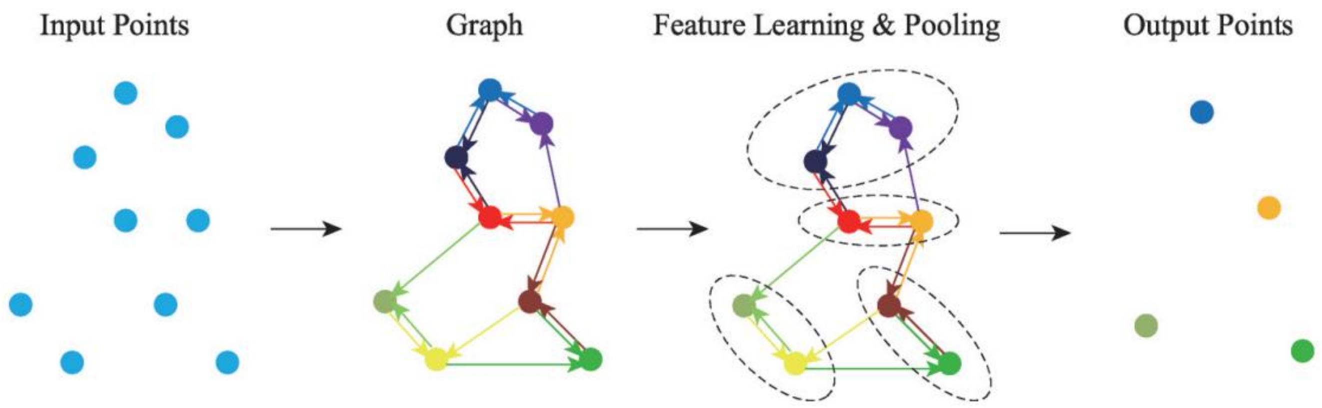
- 단점:
  - Point 마다 연산이 필요해 많은 Computation Cost 필요
  - 네트워크의 연산량이 Point 개수에 따라 영향 받아 많이 느려짐
    - Set Abstraction (SA), Feature Propagation (FP)

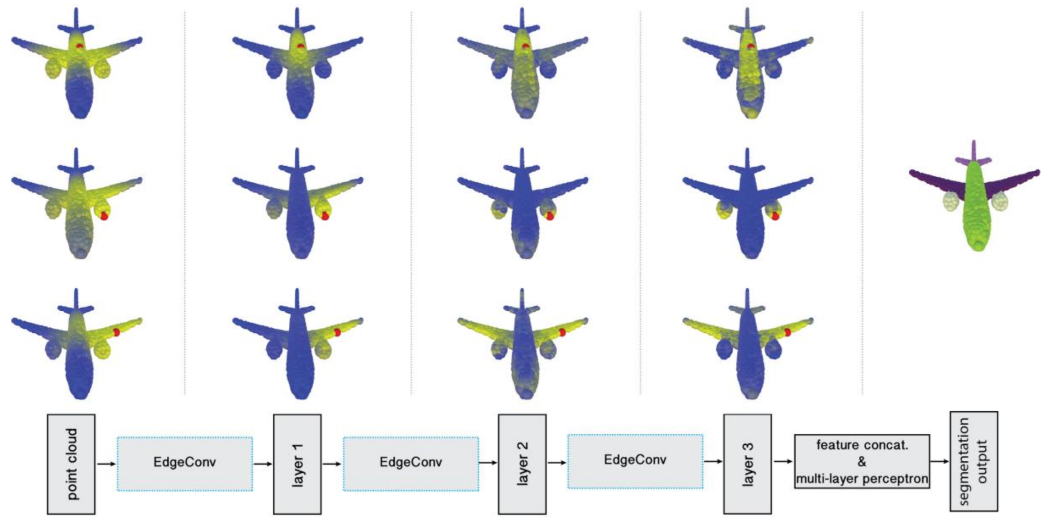
- 다른 Method에 MLP + Max Pooling의 조합을 전파

Point Cloud Representation

Graph-based

- 3D Point Cloud로 그래프를 구성해 Point(Vertex)간 관계(Edge)를 표현 (DGCNN)
- $G = (V, E)$





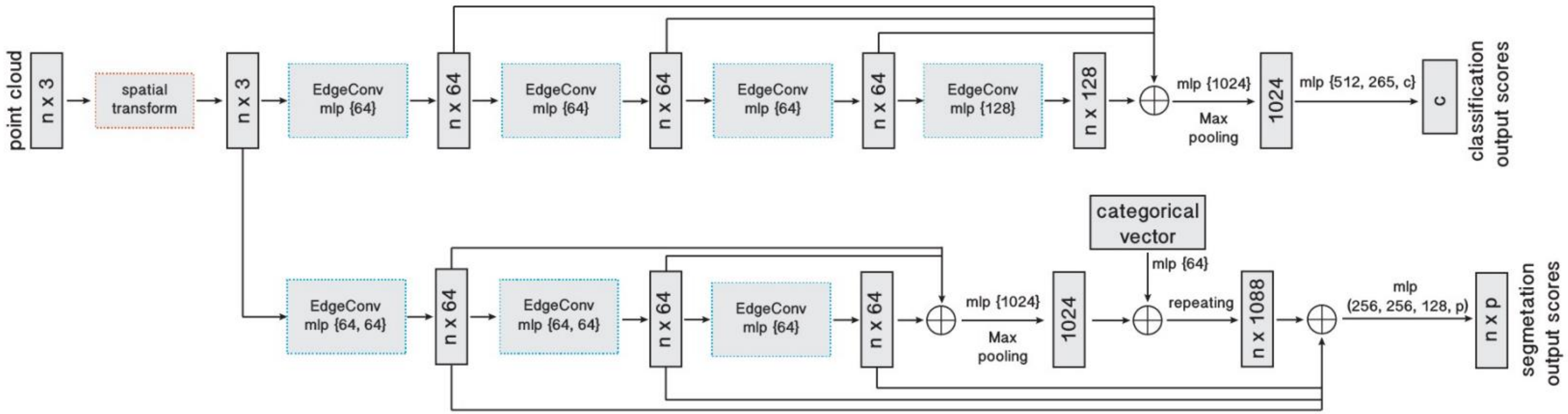
"Deep learning based point cloud registration: an overview"

"Dynamic Graph CNN for Learning on Point Clouds"

Point Cloud Representation

DGCNN

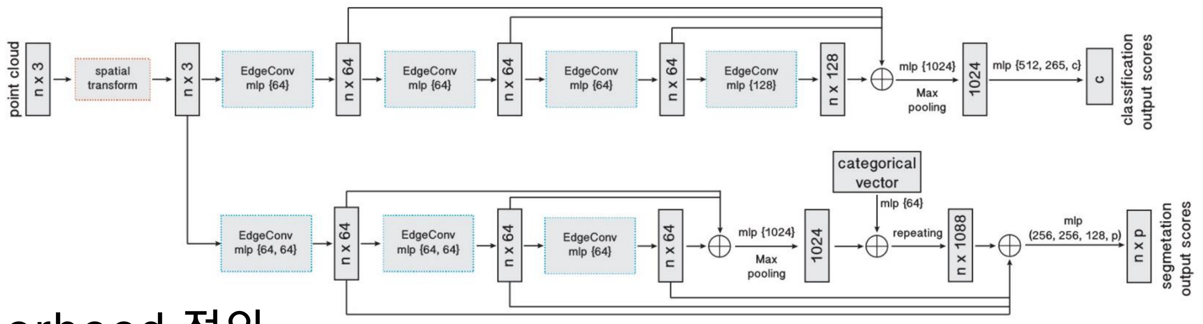
- PointNet은 Point간 관계를 설명해는 Topological Information 정보 부재
- 그래프로 Edge 정보를 학습하는 EdgeConv 제안
- PointNet과 유사 (MLP를 EdgeConv로 대체)



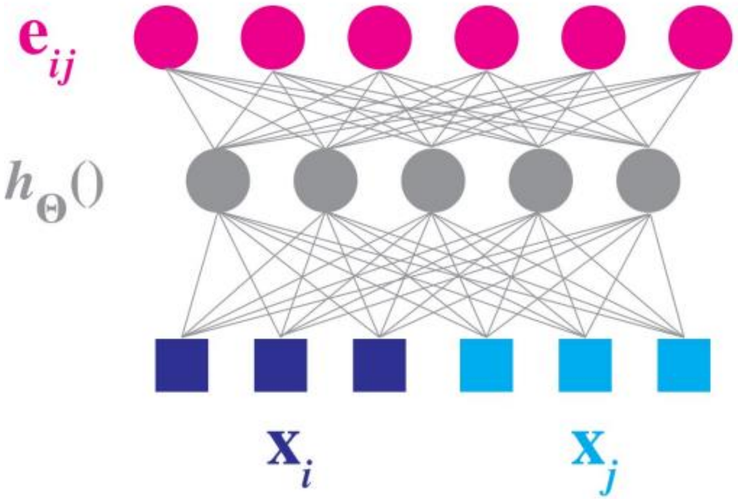
“Dynamic Graph CNN for Learning on Point Clouds”

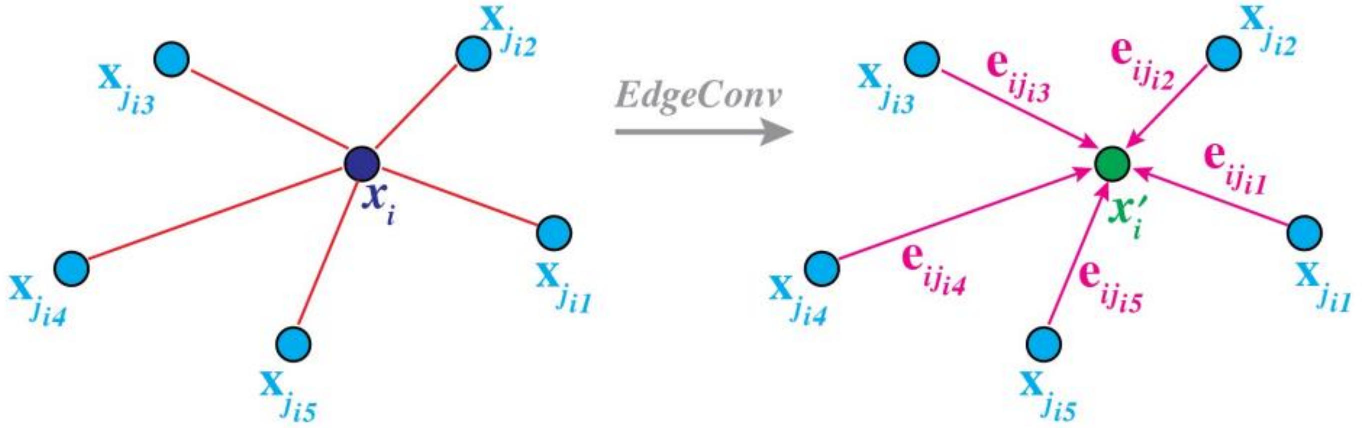
Point Cloud Representation

DGCNN



- Edge Convolution
  - 각 레이어 마다 그래프를 생성 및 업데이트
    - $N \times C \rightarrow N \times K_l \times C \rightarrow N \times C'$
  - KNN 알고리즘과 Feature Distance로 Neighborhood 정의
  - MLP와 Max Pooling 사용
    - $e'_{ijm} = \mathrm{ReLu}(\theta_m \cdot (x_j - x_i) + \phi_m \cdot x_i)$
    - $x'_i = \max(e'_{ijm})$





"Dynamic Graph CNN for Learning on Point Clouds"

Point Cloud Representation

Graph-based

- 장점:
  - Topological Information을 사용하여 Point-based보다 강화된 Representation 능력

- 단점:
  - Graph 생성 및 추가 연산으로 인해 높은 Computation cost와 메모리양 필요
  - Training & Inference 속도 느림

Applications

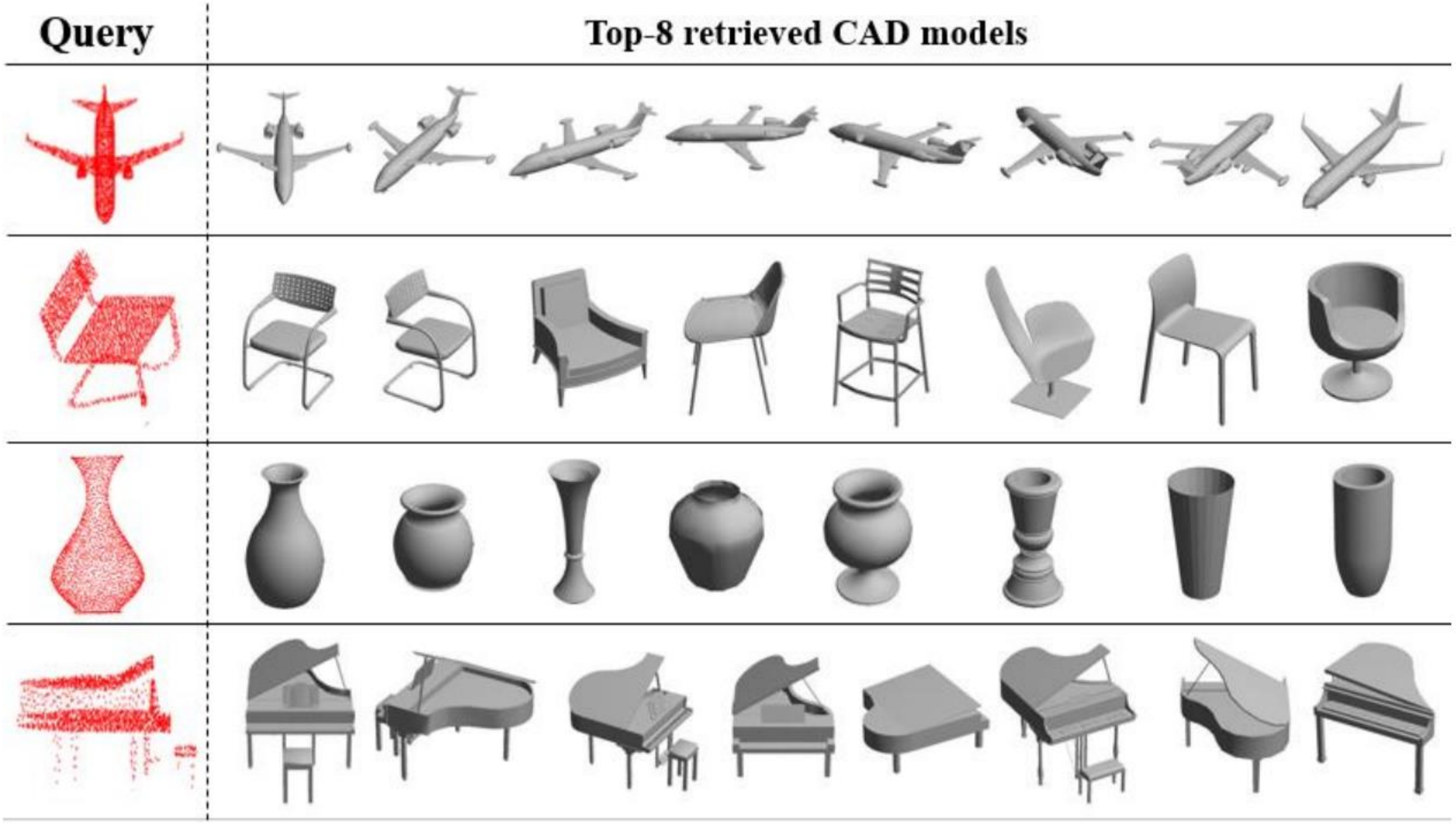
Point Cloud 딥러닝 적용 분야

- Classification & Segmentation
- Object Detection
- Registration
- 3D Shape Generation & Deformation

Applications

Classification & Segmentation

- Classification : ModelNet

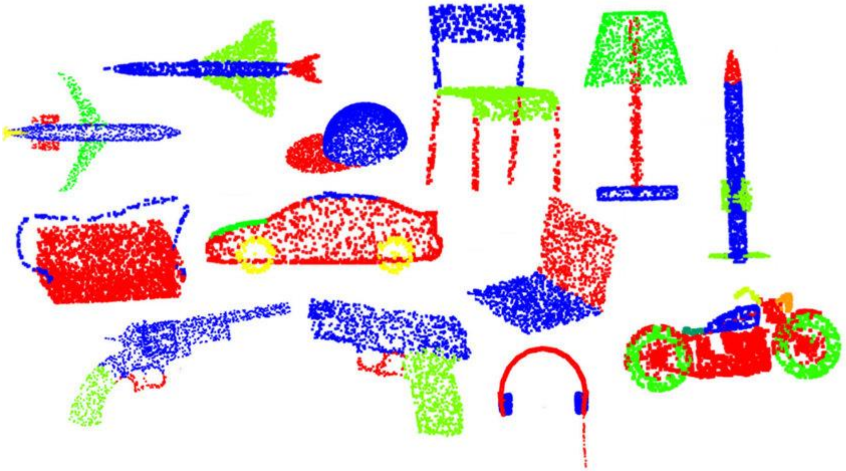


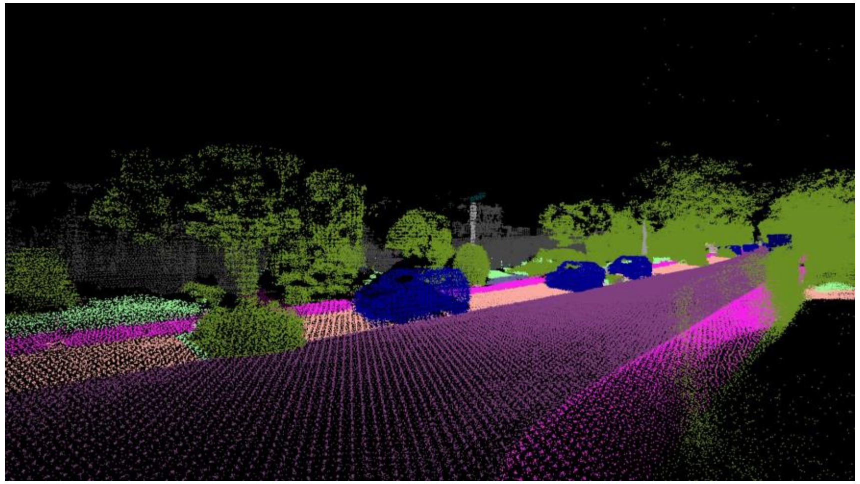
“ResSANet: Learning Geometric Information for Point Cloud Processing”

Applications

Classification & Segmentation

- Part Segmentation: ShapeNet
- Indoor Segmentation: ScanNet, SUN RGB-D, S3DIS
- Outdoor Segmentation: KITTI 360





"Local Feature Extraction Network for Point Cloud Analysis"

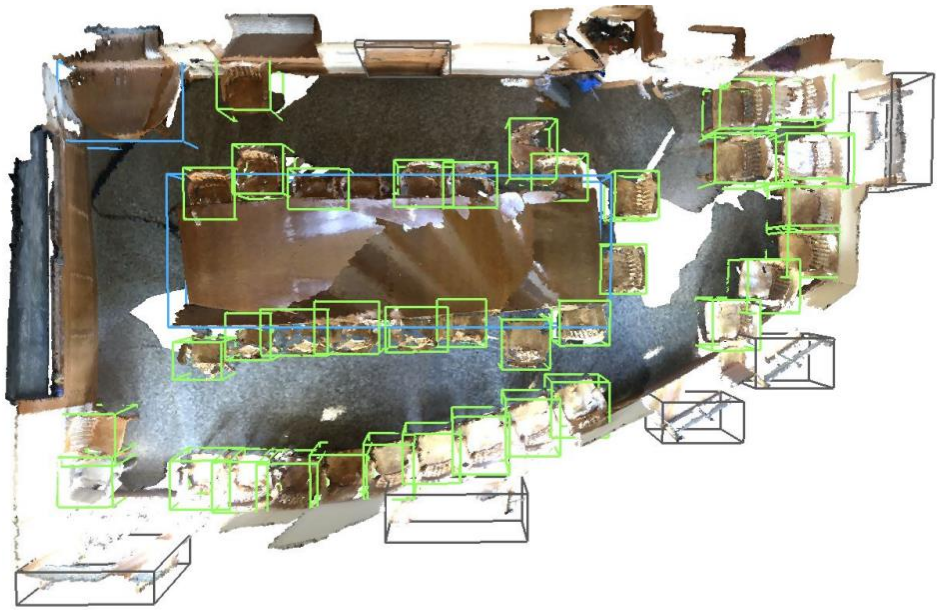
"ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes"

https://www.cvlibs.net/datasets/kitti-360/

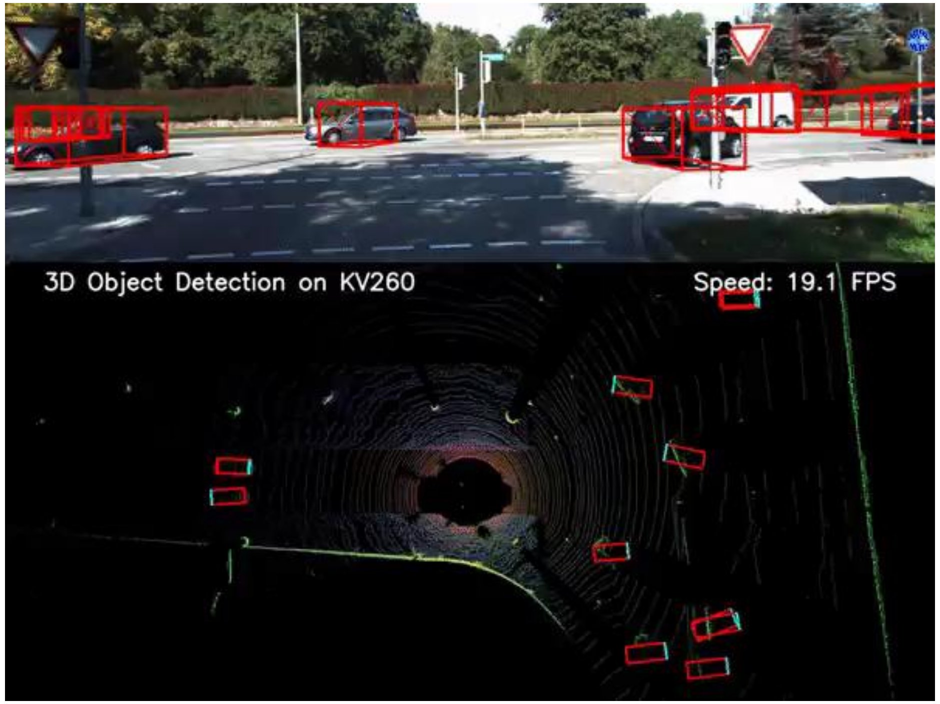
Applications

Object Detection

- Indoor : SUN RGB-D, ScanNet
- Outdoor : KITTI, Waymo Open Dataset, nuScenes



“Deep Hough Voting for 3D Object Detection in Point Clouds”

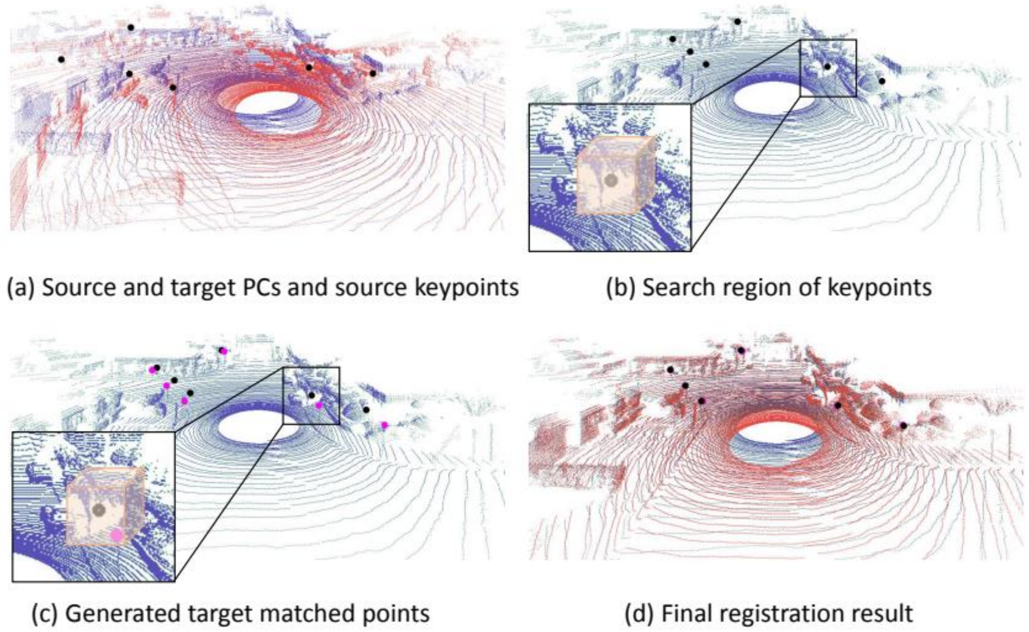


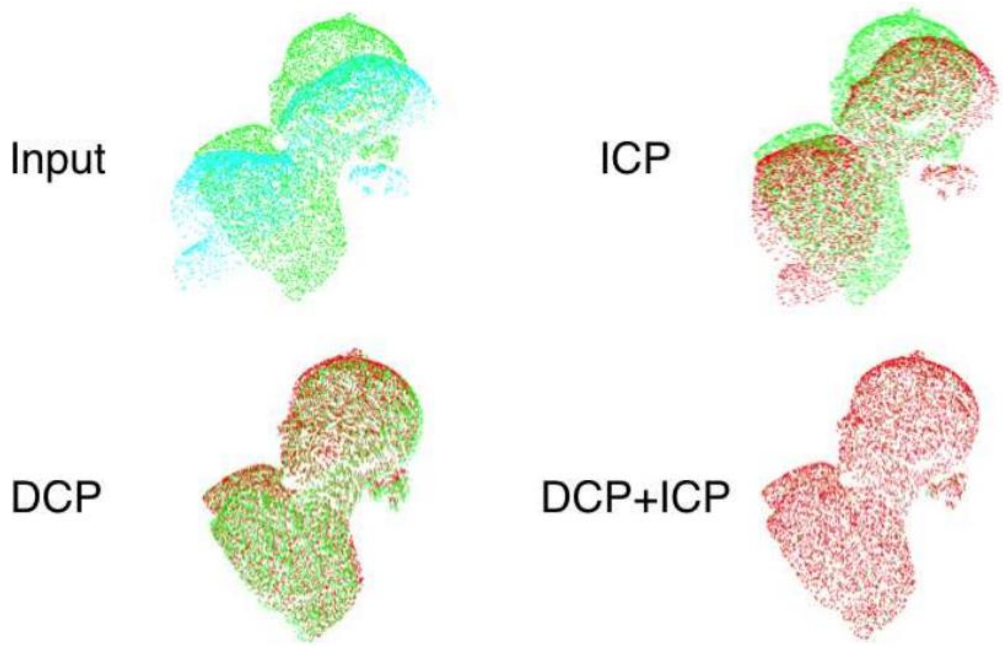
https://www.hackster.io/447969/point-clouds-based-3d-object-detection-on-pynq-dpu-c26930

Applications

Registration

- Odometry: KITTI Odometry
- Registration: ModelNet, 3D Match





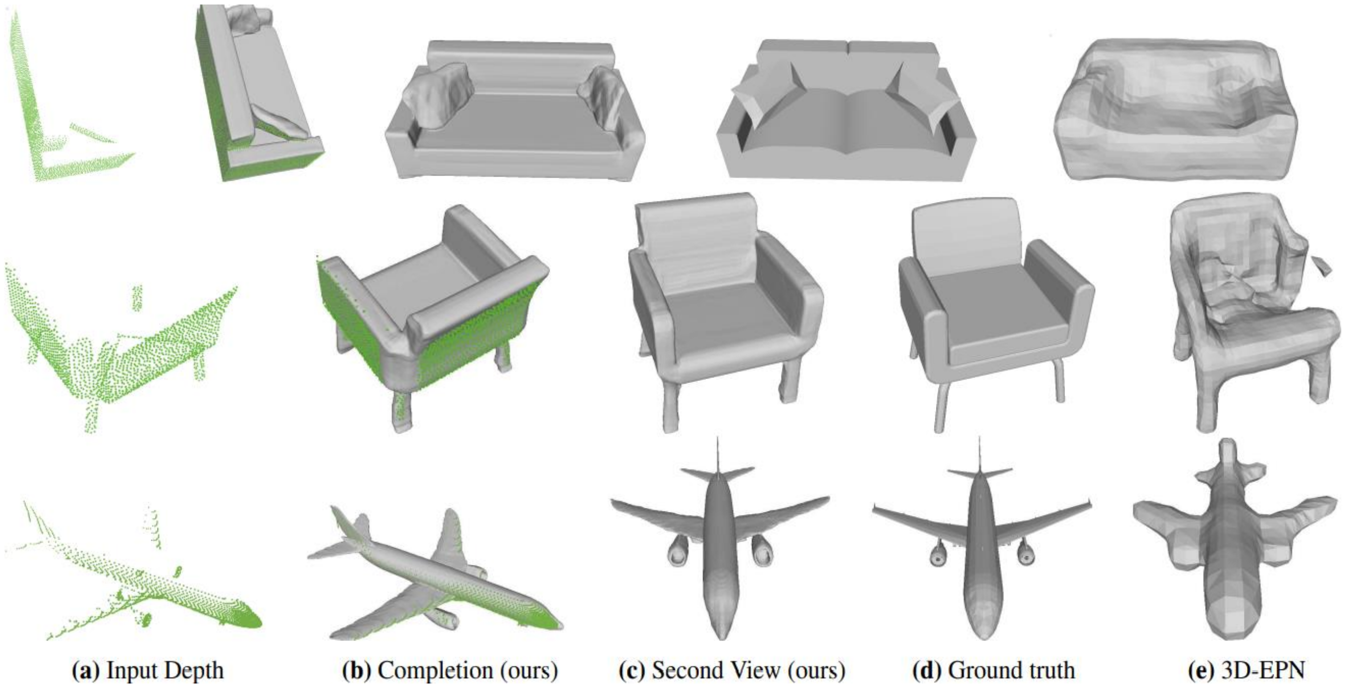
“DeepVCP: An End-to-End Deep Neural Network for Point Cloud Registration”

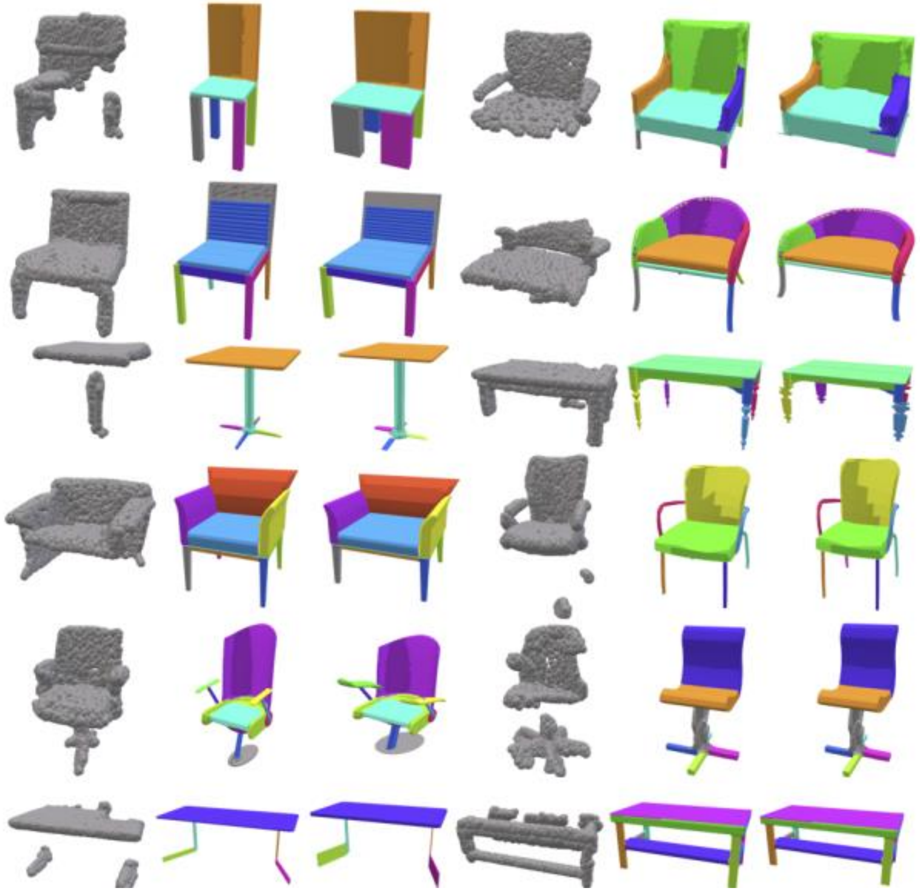
“Deep Closest Point: Learning Representations for Point Cloud Registration”

Applications

3D Shape Generation & Deformation

- Dataset: ShapeNet





"DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation"

"Joint Learning of 3D Shape Retrieval and Deformation"

Thank You

[3D Sensor Data Processing Curriculum]