
Point Transformer

2022.2.6

박진혁

목차

1. Introduction
2. Architecture of Point Transformer
3. Experiment
4. Conclusion

Introduction

❖ What is Point transformer?

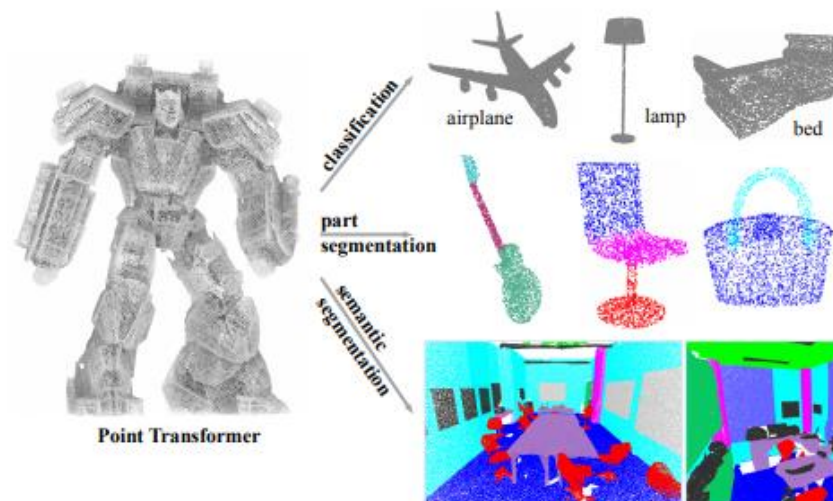
- 2022년 2월 5일 기준 136회 인용
- Point cloud classification과 segmentation을 Transformer에 적용한 논문

Point Transformer

Hengshuang Zhao^{1,2} Li Jiang³ Jiaya Jia³ Philip Torr¹ Vladlen Koltun⁴

¹University of Oxford ²The University of Hong Kong

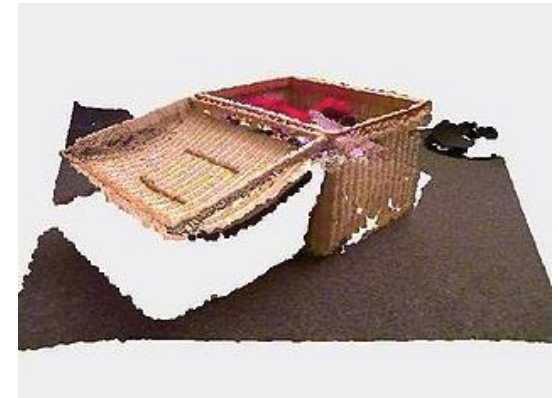
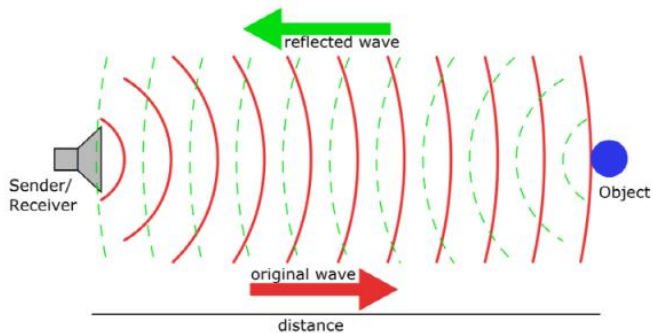
³The Chinese University of Hong Kong ⁴Intel Labs



Introduction

❖ Point cloud

- Lidar 센서, RGB-D센서 등으로 수집되는 데이터
- 3차원 공간상에 퍼져있는 여러 포인트(point)의 집합(set cloud)를 의미
- 2D 이미지와 달리 z축 정보를 가지고 있기 때문에 N*3 Numpy배열로 표현됨

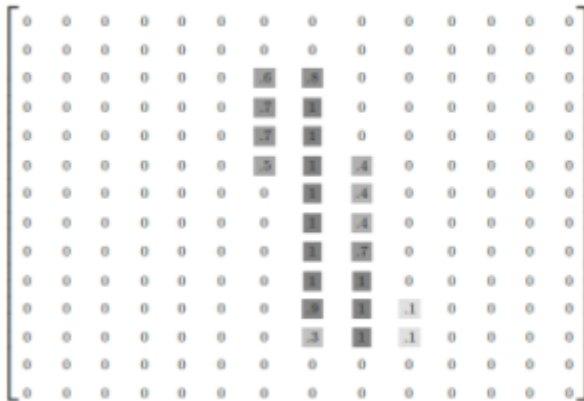


<Sensor로 수집된 point cloud>

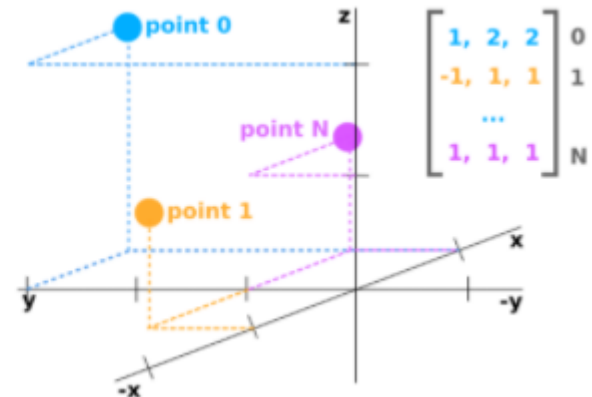
Introduction

❖ Point cloud

- Lidar 센서, RGB-D센서 등으로 수집되는 데이터
- 3차원 공간상에 퍼져있는 여러 포인트(point)의 집합(set cloud)를 의미
- 2D 이미지와 달리 z축 정보를 가지고 있기 때문에 N*3 Numpy배열로 표현됨



<이미지 데이터>



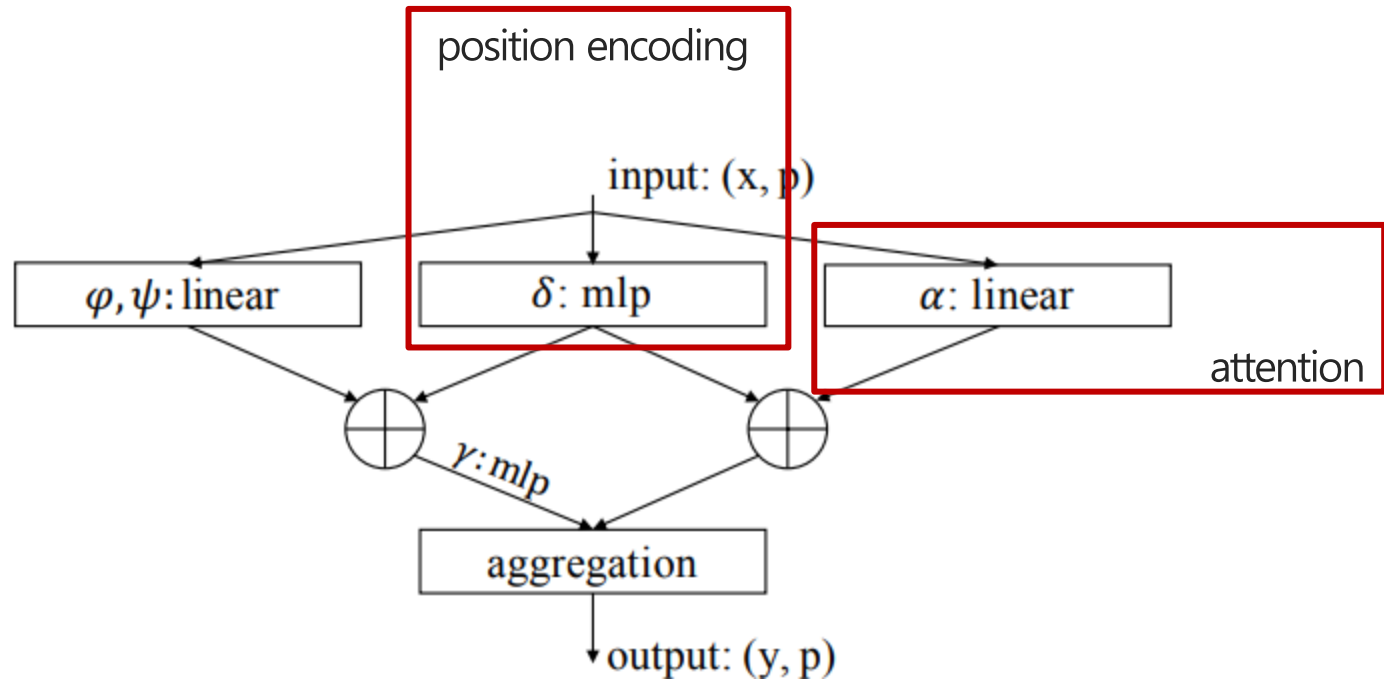
<Point cloud 데이터>

Architecture of Point Transformer

$$\mathbf{y}_i = \sum_{\mathbf{x}_j \in \mathcal{X}(i)} \rho(\gamma(\varphi(\mathbf{x}_i) - \psi(\mathbf{x}_j) + \delta)) \odot (\alpha(\mathbf{x}_j) + \delta)$$

❖ Point Transformer Layer

- 기준점과 인접한 k개 점들의 feature vector를 얻어 기준점과 인접점의 차이를 구함
- 기준점과 인접점의 차이에 대한 vector는 attention score와 곱해짐
- 최종 적으로는 attention-weighted feature vector를 더해서 최종 vector를 계산함

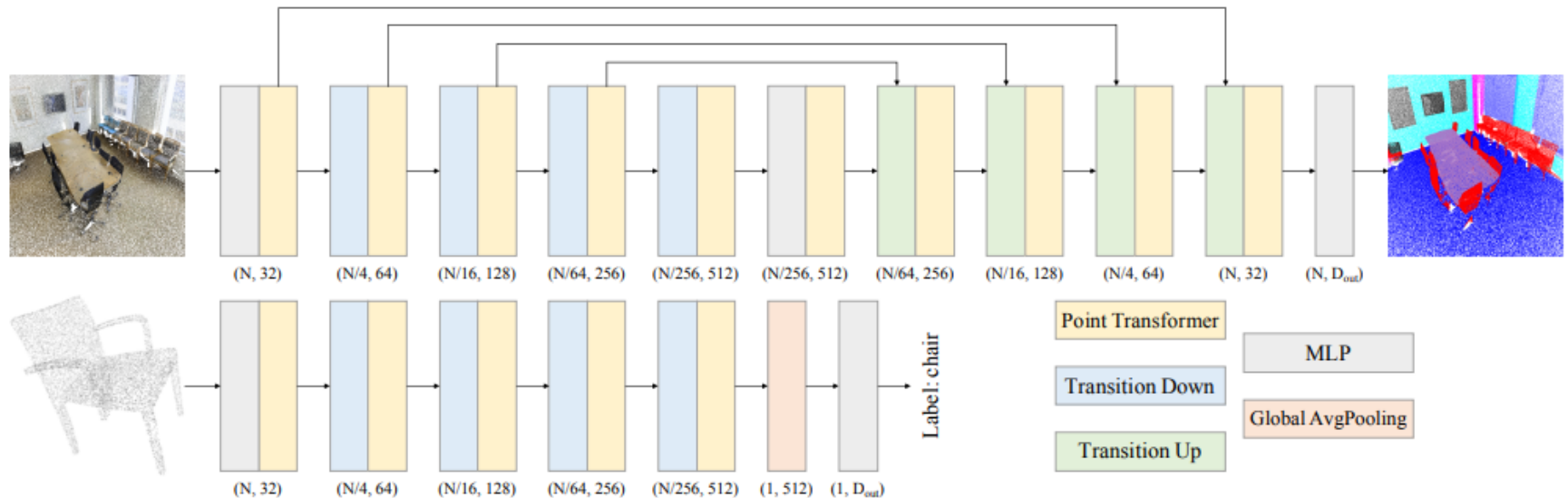


<Point transformer layer>

Architecture of Point Transformer

❖ Architecture of Point Transformer

- Semantic segmentation과 classification 진행
- U-Net구조와 유사함



<Point transformer networks for semantic segmentation (top) and classification (bottom)>

Architecture of Point Transformer

❖ Positional encoding

- Positional encoding vector
 - x,y,z 좌표값을 2개의 linear layer와 하나의 ReLU로 구성된 MLP로 얻음
 - Self-attention 중간 positional encoding vector를 추가하여 점들간의 위치 구조 반영

$$\delta = \theta(\mathbf{p}_i - \mathbf{p}_j)$$

δ, θ : position encoding function

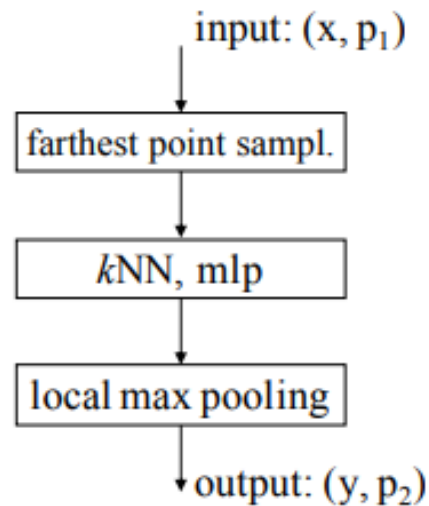
\mathbf{p}_i : 3D point coordinate for point i

\mathbf{p}_j : 3D point coordinate for point j

Architecture of Point Transformer

❖ Transition down

- Point set의 크기를 줄여줌
 - Farthest point sampling 알고리즘을 사용
 - 서로 간의 거리가 가장 먼 점을 선택하는 방법론
- Sampling을 통해 얻은 vector에서 local하게 maxpooling 진행

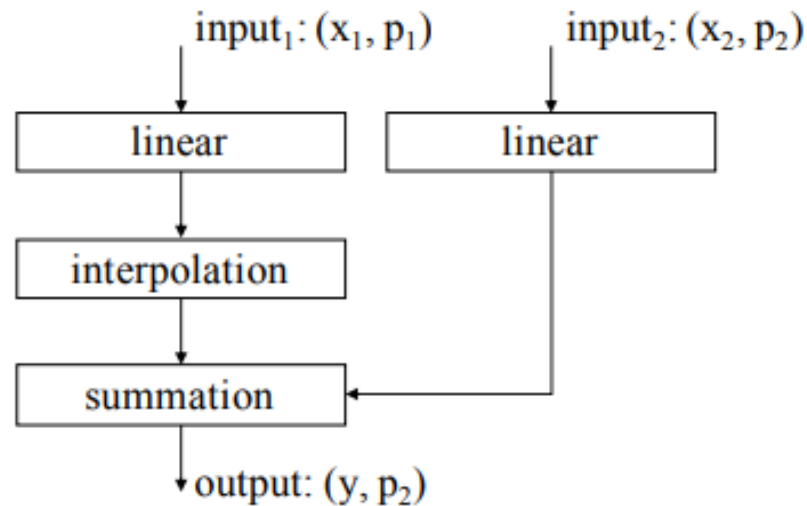


<transition down>

Architecture of Point Transformer

❖ Transition up

- Point set의 크기를 이전의 크기로 키워줌
- Segmentation task에서는 모든 pixel에 대한 label을 얻어야 하기 때문에 필수적임
- Interpolation block과 skip-connection으로 구성됨



<transition up>

Experiment

❖ Sematic segmentation

- S3DIS dataset을 사용하여 평가
 - 실내에 대한 segmentation 성능 평가
 - 다른 방법론들과 비교하여 SOTA 성능을 보여줌

Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [25]	–	49.0	41.1	88.8	97.3	69.8	0.1	3.9	46.3	10.8	59.0	52.6	5.9	40.3	26.4	33.2
SegCloud [36]	–	57.4	48.9	90.1	96.1	69.9	0.0	18.4	38.4	23.1	70.4	75.9	40.9	58.4	13.0	41.6
TangentConv [35]	–	62.2	52.6	90.5	97.7	74.0	0.0	20.7	39.0	31.3	77.5	69.4	57.3	38.5	48.8	39.8
PointCNN [20]	85.9	63.9	57.3	92.3	98.2	79.4	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	56.7
SPGraph [15]	86.4	66.5	58.0	89.4	96.9	78.1	0.0	42.8	48.9	61.6	84.7	75.4	69.8	52.6	2.1	52.2
PCCN [42]	–	67.0	58.3	92.3	96.2	75.9	0.3	6.0	69.5	63.5	66.9	65.6	47.3	68.9	59.1	46.2
PAT [50]	–	70.8	60.1	93.0	98.5	72.3	1.0	41.5	85.1	38.2	57.7	83.6	48.1	67.0	61.3	33.6
PointWeb [55]	87.0	66.6	60.3	92.0	98.5	79.4	0.0	21.1	59.7	34.8	76.3	88.3	46.9	69.3	64.9	52.5
HPEIN [13]	87.2	68.3	61.9	91.5	98.2	81.4	0.0	23.3	65.3	40.0	75.5	87.7	58.5	67.8	65.6	49.4
MinkowskiNet [37]	–	71.7	65.4	91.8	98.7	86.2	0.0	34.1	48.9	62.4	81.6	89.8	47.2	74.9	74.4	58.6
KPConv [37]	–	72.8	67.1	92.8	97.3	82.4	0.0	23.9	58.0	69.0	81.5	91.0	75.4	75.3	66.7	58.9
PointTransformer	90.8	76.5	70.4	94.0	98.5	86.3	0.0	38.0	63.4	74.3	89.1	82.4	74.3	80.2	76.0	59.3

Experiment

❖ Classification

- ModelNet40 dataset을 사용하여 평가
 - 물체의 shape classification 성능을 평가
 - 다른 방법론들과 비교하여 SOTA 성능을 보여줌

Method	input	mAcc	OA
3DShapeNets [47]	voxel	77.3	84.7
VoxNet [23]	voxel	83.0	85.9
Subvolume [26]	voxel	86.0	89.2
MVCNN [34]	image	–	90.1
PointNet [25]	point	86.2	89.2
A-SCN [48]	point	87.6	90.0
Set Transformer [17]	point	–	90.4
PAT [50]	point	–	91.7
PointNet++ [27]	point	–	91.9
SpecGCN [40]	point	–	92.1
PointCNN [20]	point	88.1	92.2
DGCNN [44]	point	90.2	92.2
PointWeb [55]	point	89.4	92.3
SpiderCNN [49]	point	–	92.4
PointConv [46]	point	–	92.5
Point2Sequence [21]	point	90.4	92.6
KPConv [37]	point	–	92.9
InterpCNN [22]	point	–	93.0
PointTransformer	point	90.6	93.7

Experiment

❖ Ablation studies

- K의 개수에 따른 실험 진행
 - K가 4,8인 경우 주변 구조 관계에 대한 정보 부족으로 학습성능이 저하됨
 - K가 32, 64인 경우 너무 많은 연관성 없는 점들이 noise로 작용하여 학습성능이 저하됨

k	mIoU	mAcc	OA
4	59.6	66.0	86.0
8	67.7	73.8	89.9
16	70.4	76.5	90.8
32	68.3	75.0	89.8
64	67.7	74.1	89.9

<Number of neighbors k in the definition of local neighborhoods>

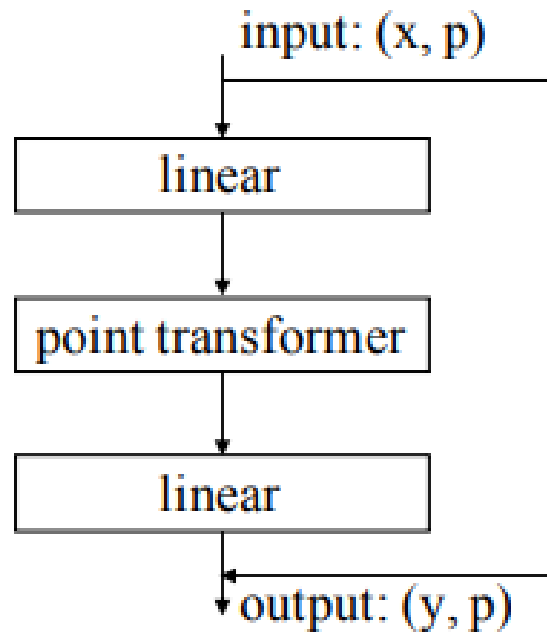
Conclusion

❖ Summary

- Point cloud를 transformer 구조에 적용한 논문
- Segmentation task와 classification task에 적용 가능한 방법론
- Transformer와 UNet과 같은 모델 구조를 사용한 방법론
- Transformer에 적용한 초창기 논문이지만 다른 방법론에 비해 우수한 성능을 보임
- Graph알고리즘을 transformer에 적용한 초창기 연구이므로 향후 발전가능성이 존재함

Appendix

❖ Point transformer block



<point transformer block>