Generalizing discrete convolutions for unstructured point clouds

Eurographics 3DOR 2019

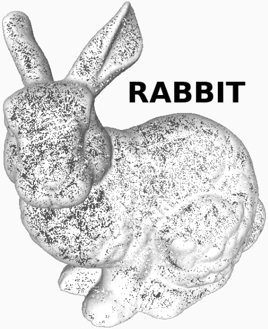
Alexandre Boulch
www.boulch.eu

Image Vision Machine Learning team

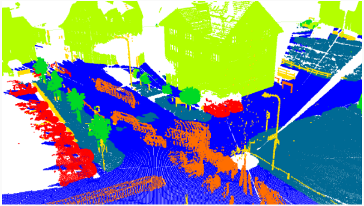
Image, Vision and Machine Learning team at ONERA :

- Machine learning
- 2D - 3D scene understanding
- Vision based robotics
- Remote sensing
- Metrology

Objective

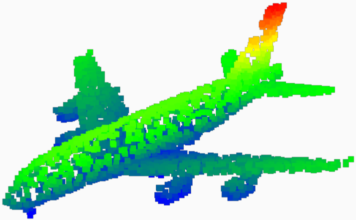


Classification



Semantic segmentation

Point clouds



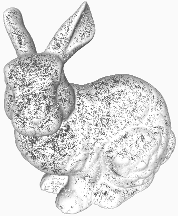
Point cloud P : set of points $p_i, i \in [1, ..., |P|]$.

A point cloud is

- unstructured : not sampled on grid
- unordered : invariant by permutation of points
- scale less : e.g. CAD, photogrammetry
- defined by point coordinates only

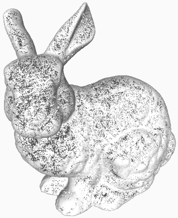
Previous works

Using machine learning on point clouds

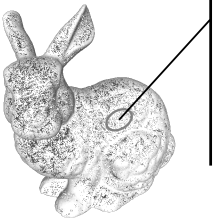


Pre-processing: representation → **Machine learning** → Rabbit

Pre-processing : hand crafted features



Hand crafted features → **Machine learning** → Rabbit

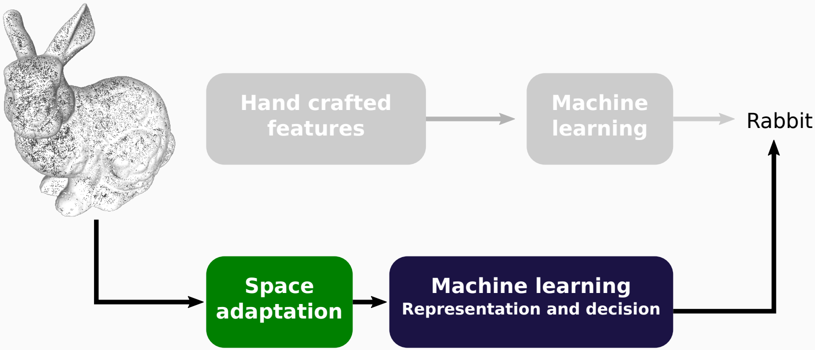


[HWS16]
Verticality
Planarity
Anisotropy
...

[JH99]
Spin images

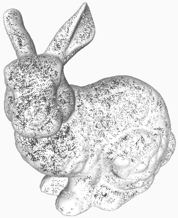
Random forests
SVM
...

Pre-processing

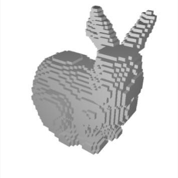


- Designing features is difficult $\rightarrow$ learn them
- Exploit images processing approaches $\rightarrow$ adapt space

Pre-processing : voxelization



Space adaptation → **Machine learning** **Representation and decision** → Rabbit



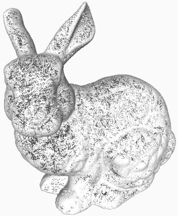
Voxels

3D CNNs
[WSK+15][MS15]

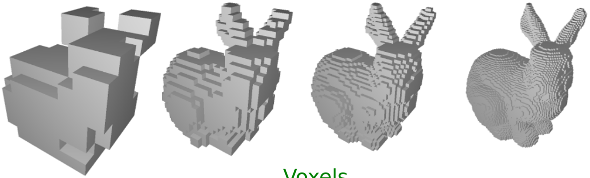
Sparse Conv.
[Gra15] [GvdM17]

Object focus
[LPS+16][WP15]

Pre-processing : voxelization



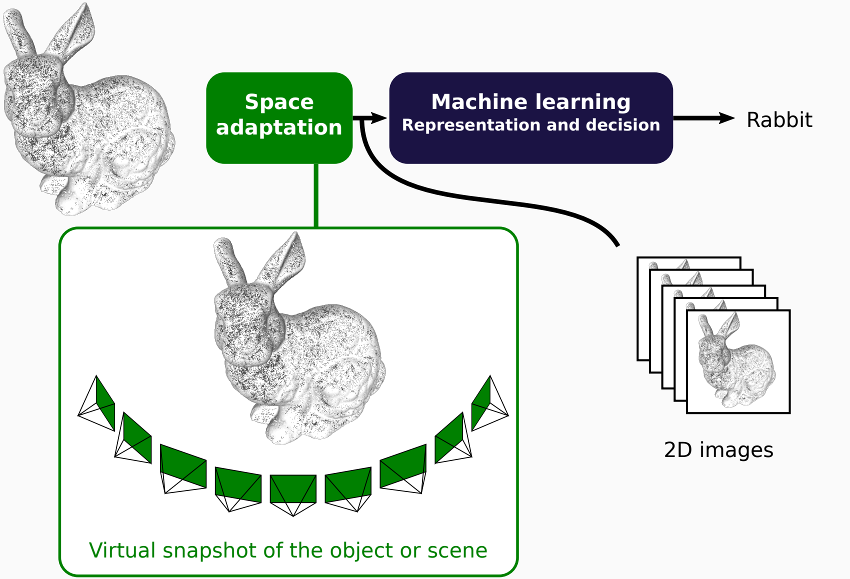
Space adaptation → **Machine learning** **Representation and decision** → Rabbit



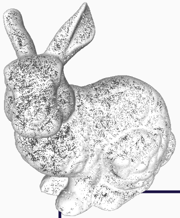
Voxels

Difficulties : Voxel sizes ? Voxel orientation ?

Pre-processing : 2D approaches

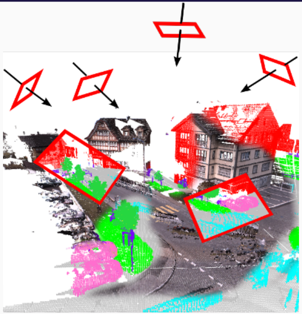


Pre-processing : 2D approaches



Space adaptation → **Machine learning** Representation and decision → Rabbit

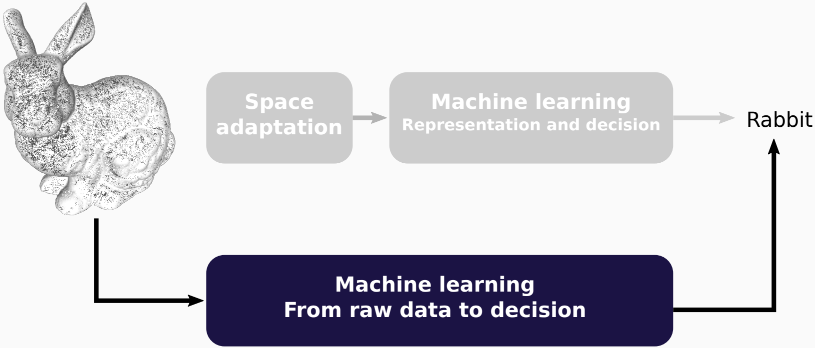
Classification: [SMLLM15]



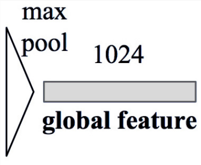
Segmentation: [BGLSA17]

Difficulties : Snapshot strategies ?

Direct point processing



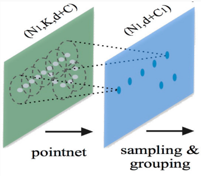
Direct point processing



PointNet : Qi et al [Qi+17a]

Permutation inv. : Max pooling.

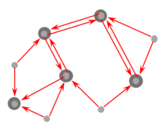
Use location as features.



PointNet++ : Qi et al [Qi+17b]

Hierarchical reprentation.

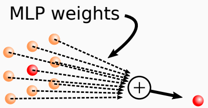
Use location as features.



PointCNN : Li et al [Li+18]

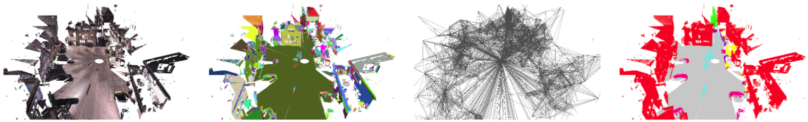
Input projection on kernel.

Use location as features.



PCCN : Wang et al [Wan+18a]

Feature weighting using MLP



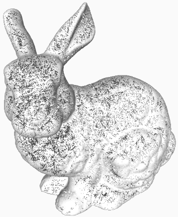
(a) RGB point cloud (b) Geometric partition (c) Superpoint graph (d) Semantic segmentation

Loic LANDRIEU et Martin SIMONOVSKY. “Large-scale point cloud semantic segmentation with superpoint graphs”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 4558-4567

Use an offline segmentation of the cloud followed by PointNet and GRU learning.

Generalizing convolutions

Objective



Machine learning → Rabbit

CNNs have proved very efficient for image / voxel processing

Objective

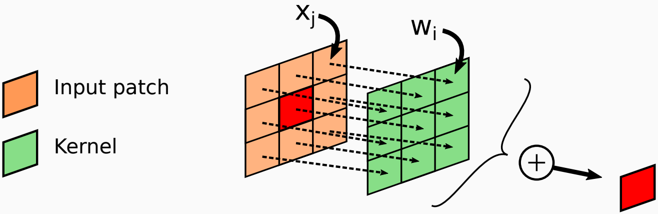
- Adapt convolutions do sparse, unstructured data
- Stick as mush as possible to the original formulation

Discrete convolutions

Formulation for images

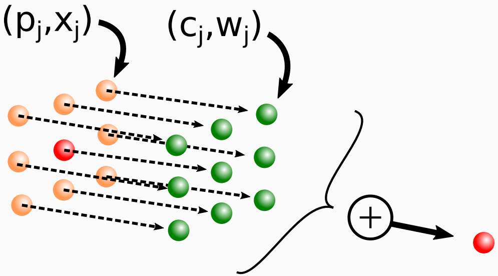
$$y = \beta + \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_i = \beta + \sum_{i=1}^{N} \sum_{j=1}^{|X|} \mathbf{w}_i \mathbf{x}_j \mathbf{1}(i,j)$$

with $\mathbf{1}(i,j)$ the indicator function.



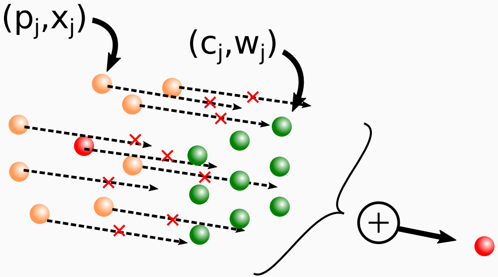
Formulation with points

$$y = \beta + \frac{1}{|X|} \sum_{j=1}^{|X|} \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_j \mathbf{1}(\mathbf{c}_i, \mathbf{p}_j)$$



Valid for structured inputs.

Point cloud particularities



Unstructured inputs would lead to zero value almost all the time.

Indicator function is not the right function for unstructured inputs.

Continuous convolution

$$y = \beta + \frac{1}{|X|} \sum_{j=1}^{|X|} \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_j \phi_i(p_j, C)$$

- Interpolation to neighbors

Somehow suppose a grid for the kernel.

Hang Su et al. "SPLATNet : Sparse Lattice Networks for Point Cloud Processing". In : *arXiv preprint arXiv :1802.08275* (2018)

Continuous convolution

$$y = \beta + \frac{1}{|X|} \sum_{j=1}^{|X|} \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_j \phi_i(p_j, C)$$

ϕ construction :

- dicrease with the a distance
- deal with relative positions of kernel elements

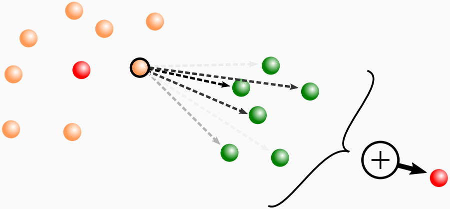
Inverse ℓ_2 distance ? Gaussian functions ? ...

How to tune parameters ?

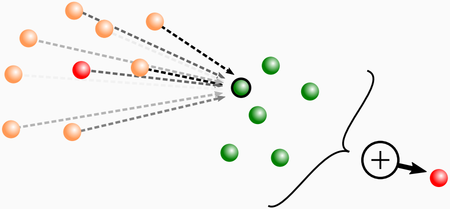
Our approach

Use a MLP for ϕ function [Li+18 ; Wan+18a] :

$$y = \beta + \frac{1}{|X|}\sum_{j=1}^{|X|}\sum_{i=1}^{N}\mathbf{w}_i\mathbf{x}_j MLP_i(p_j, C)$$



Each input influences
each kernel elements



Each kernel element sees
the whole input

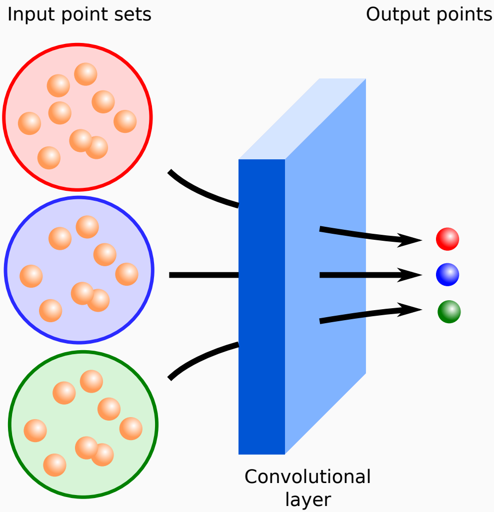
Our approach

Use a MLP for ϕ function [Li+18 ; Wan+18a] :

$$y = \beta + \frac{1}{|X|} \sum_{j=1}^{|X|} \sum_{i=1}^{N} \mathbf{w}_i \mathbf{x}_j MLP_i(p_j, C)$$

Properties

- Permutation invariant
- Translation invariant
- Low sensibility to input size
- Low sensibility to input scale if input normalized to unit ball



Input

Point sets (neighborhoods) : locations + features

Output

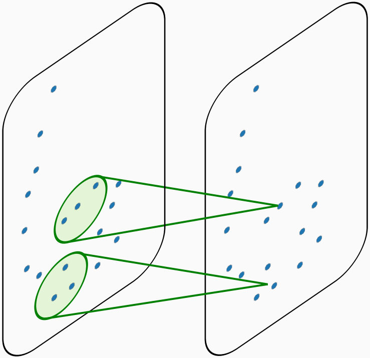
Features at given output locations (center of neighborhoods)

No dimension reduction

One neighborhood for each point of the input cloud.

Images :

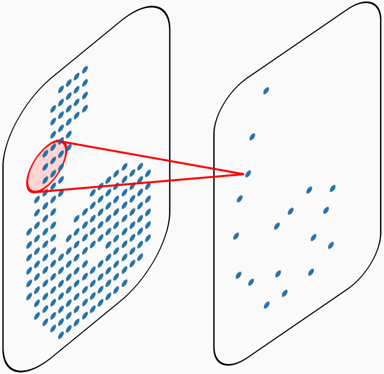
Convolution with stride 1.



Number of neighborhoods lower than input cloud size.

Images :

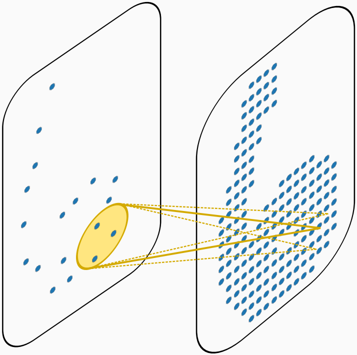
Convolution with stride ≥ 2.



Output point cloud size greater than input cloud size.

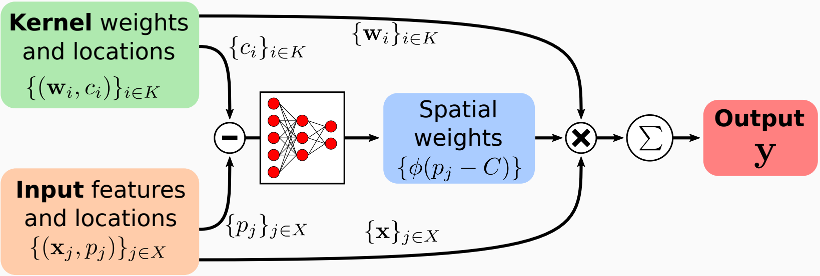
Images :

Convolution transpose.



Implementation

Block representations



Practical implementation

Python implementation

Only differentiable operations Autograd usage (PyTorch or Tensorflow)

```
# Neighborhoods
features = input.view(-1, input.size(2))[indices]
pts = points.view(-1, points.size(2))[indices]

# Relative positions
pts = pts - next_points.unsqueeze(2)

# Normalization
if normalize:
    maxi = torch.sqrt((pts**2).sum(3).max(2)[0])
    maxi[maxi==0] = 1
    pts = pts / maxi.view(maxi.size()+(1,1,))

# wieghting MLP
dists = pts.view(pts.size()+(1,)) - centers
dists = dists.view(dists.size(0), dists.size(1),
                   dists.size(2), -1)
dists = F.relu(l1(dists))
dists = F.relu(l2(dists))
dists = F.relu(l3(dists))
dists = dists.unsqueeze(3)

# Features
features = features.view(features.size()+(1,)) * dists
features = features.mean(2)
features = features.view(features.size()+(1,)) * weight
features = features.sum([2,3])

# Bias
features = features + bias
```

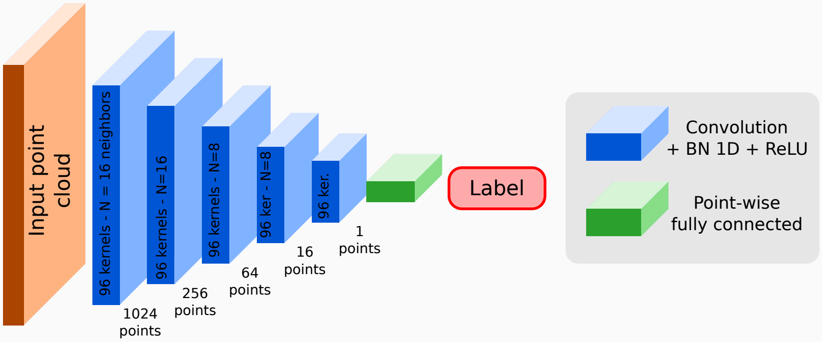
Neighborhoods computation

Computed using search trees from Scikit-learn.

Precomputation for efficiency, all neighborhoods are computed in the data loader.

Experiments

Network for classification



Classification : number of sampling

MNIST dataset	
1 sampling	99.55
8 samplings	99.59
16 samplings	99.61

ModelNet40 dataset		
1 sampling	90.1	86.8
8 samplings	91.2	87.8
16 samplings	91.6	88.1

Number of runs

Random dimension reduction induce variable results. Averaging the results leads to better results.

Similar to crops for image classification.

Classification : state of the art

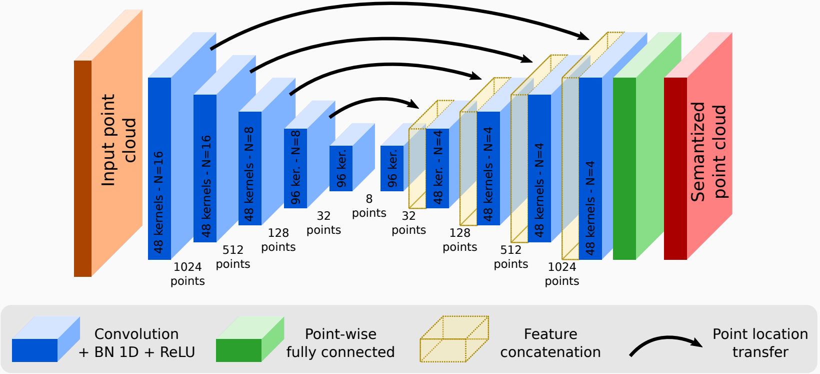
MNIST dataset

Methods	OA
NiN [LCY13]	99.53
PointNet++ [Qi+17b]	99.49
PointCNN [Li+18]	99.54
Ours	
16 samplings	**99.61**

ModelNet40 dataset

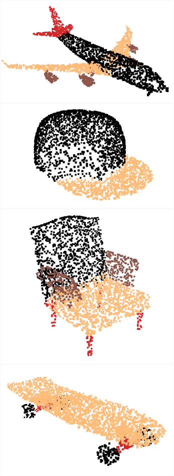
Methods	OA	AA
DGCNN [Wan+18c]	92.2	**90.2**
PointNet++ [Qi+17b]	90.7	
PointCNN	**92.2**	88.1
Ours		
16 samplings	91.6	88.1

Network for segmentation

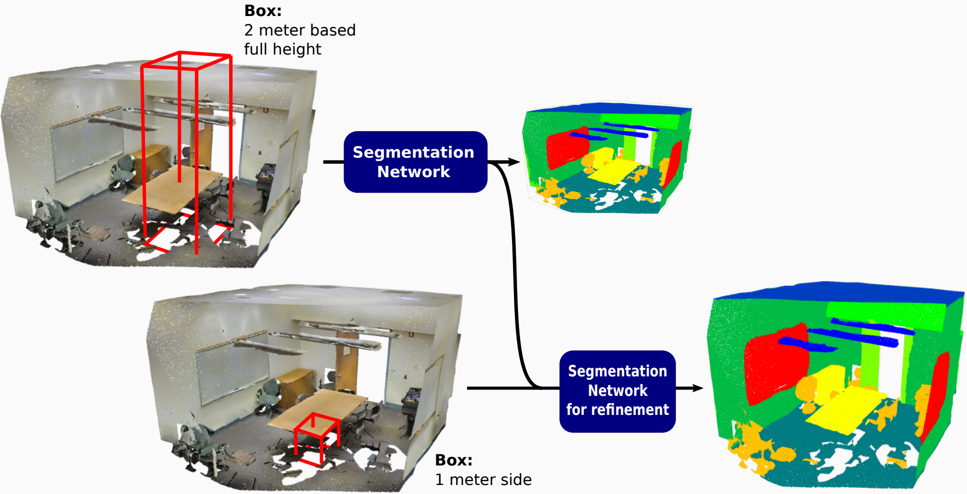


Part segmentation

Method	pIoU	mpIoU
SPLATNet [Su+18]	85.4	83.7
DGCNN [Wan+18c]	85.1	82.3
PointNet [Qi+17a]	83.7	80.4
PointNet++ [Qi+17b]	85.1	81.9
SGPN [Wan+18b]	85.8	82.8
PointCNN [Li+18]	86.14	**84.6**
Ours 1024 pts		
16 trees	**93.1**	82.6



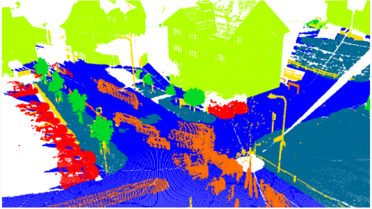
Indoor segmentation : S3DIS



Indoor segmentation : S3DIS

Method	OA	mAcc	mIoU
PointNet [Qi+17a]	78.5	66.2	47.6
SPGraph [LS18]	85.5	73.0	62.1
RSNet [HWN18]	-	66.45	56.47
PCCN [Wan+18a]	-	67.01	58.27
PointCNN [Li+18]	**88.14**	**75.61**	**65.39**
Ours			
1 scale 2m	84.05	-	55.36
2 scales 2m,1m	84.93	-	58.54

Outdoor, large scale segmentation : Semantic8

Method	AvIoU	OA
TML-PC [MZ+14]	0.391	0.745
TMLC-MS [HWS16]	0.494	0.850
PointNet++ [Qi+17b]	0.631	0.857
SnapNet [Bou+17]	0.674	0.910
SPGraph [LS18]	**0.762**	**0.929**
Ours	0.666	0.898
ranking	*3*	*3*

Perspectives and conclusions

Perspectives

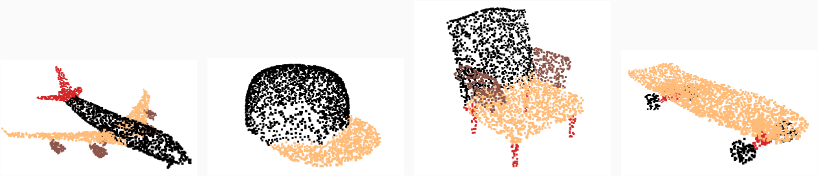
- Work on architecture design
- Training strategy : multiscale, layer initialization . . .
- Extend layers

Conclusion

- Competitive results
- A single architecture on all datasets
- Trained on a 12G NVidia Titan GPU

Code available

`https://github.com/aboulch/ConvPoint`



[AML18] Matan Atzmon, Haggai Maron et Yaron Lipman. “Point Convolutional Neural Networks by Extension Operators”. In : *arXiv preprint arXiv :1803.10091* (2018).

[ASC11] Mathieu Aubry, Ulrich Schlickewei et Daniel Cremers. “The wave kernel signature : A quantum mechanical approach to shape analysis”. In : *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE. 2011, p. 1626-1633.

[BK10] Michael M Bronstein et Iasonas Kokkinos. “Scale-invariant heat kernel signatures for non-rigid shape recognition”. In : *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, p. 1704-1711.

[BM12] Alexandre Boulch et Renaud Marlet. “Fast and robust normal estimation for point clouds with sharp features”. In : *Computer graphics forum*. T. 31. 5. Wiley Online Library. 2012, p. 1765-1774.

[Bou+17] Alexandre Boulch et al. “SnapNet : 3D point cloud semantic labeling with 2D deep segmentation networks”. In : *Computers & Graphics* (2017).

[BSLF17] Yizhak Ben-Shabat, Michael Lindenbaum et Anath Fischer. “3D Point Cloud Classification and Segmentation using 3D Modified Fisher Vector Representation for Convolutional Neural Networks”. In : *arXiv preprint arXiv :1711.08241* (2017).

[Dai+17] Jifeng Dai et al. “Deformable convolutional networks”. In : *CoRR, abs/1703.06211* 1.2 (2017), p. 3.

[GEM18] Benjamin Graham, Martin Engelcke et Laurens van der Maaten. "3D semantic segmentation with submanifold sparse convolutional networks". In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 9224-9232.

[GM17] Benjamin Graham et Laurens van der Maaten. "Submanifold Sparse Convolutional Networks". In : *arXiv preprint arXiv :1706.01307* (2017).

[Gra15] Ben Graham. "Sparse 3D convolutional neural networks". In : *arXiv preprint arXiv :1505.02890* (2015).

[Hac+17] T. Hackel et al. "Smeantic3D.net : A new large scale point cloud classification benchmark". In : *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. T. IV-1-W1. 2017, p. 91-98.

[Hop+92] Hugues Hoppe et al. *Surface reconstruction from unorganized points*. T. 26. 2. ACM, 1992.

[HWN18] Qiangui Huang, Weiyue Wang et Ulrich Neumann. "Recurrent Slice Networks for 3D Segmentation of Point Clouds". In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 2626-2635.

[HWS16] Timo Hackel, Jan D Wegner et Konrad Schindler. "Fast semantic segmentation of 3D point clouds with strongly varying density". In : *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 3.3 (2016).

[JH99] Andrew E. Johnson et Martial Hebert. "Using spin images for efficient object recognition in cluttered 3D scenes". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), p. 433-449.

[KL17] Roman Klokov et Victor Lempitsky. “Escape from cells : Deep kd-networks for the recognition of 3D point cloud models”. In : *IEEE International Conference on Computer Vision (ICCV)*. IEEE. 2017, p. 863-872.

[LCHL18] Jiaxin Li, Ben M Chen et Gim Hee Lee. “So-net : Self-organizing network for point cloud analysis”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 9397-9406.

[LCY13] Min Lin, Qiang Chen et Shuicheng Yan. “Network in network”. In : *arXiv preprint arXiv :1312.4400* (2013).

[LeC+98] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (1998), p. 2278-2324.

[Li+16] Yangyan Li et al. “FPNN : Field probing neural networks for 3D data”. In : *Advances in Neural Information Processing Systems*. 2016, p. 307-315.

[Li+18] Yangyan Li et al. “PointCNN : Convolution On X-Transformed Points”. In : *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, p. 828-838.

[LJ07] Haibin Ling et David W Jacobs. “Shape classification using the inner-distance”. In : *IEEE transactions on pattern analysis and machine intelligence* 29.2 (2007), p. 286-299.

[LS18] Loic Landrieu et Martin Simonovsky. “Large-scale point cloud semantic segmentation with superpoint graphs”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 4558-4567.

ONERA THE FRENCH AEROSPACE LAB

[MS15] Daniel Maturana et Sebastian Scherer. “Voxnet : A 3D convolutional neural network for real-time object recognition”. In : *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE. 2015, p. 922-928.

[MZ+14] Javier A Montoya-Zegarra et al. “Mind the gap : modeling local and global context in (road) networks”. In : *German Conference on Pattern Recognition*. Springer. 2014, p. 212-223.

[Qi+16] Charles R Qi et al. “Volumetric and multi-view CNNs for object classification on 3D data”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 5648-5656.

[Qi+17a] Charles R Qi et al. “PointNet : Deep learning on point sets for 3D classification and segmentation”. In : *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* 1.2 (2017), p. 4.

[Qi+17b] Charles Ruizhongtai Qi et al. “Pointnet++ : Deep hierarchical feature learning on point sets in a metric space”. In : *Advances in Neural Information Processing Systems*. 2017, p. 5105-5114.

[RFB15] Olaf Ronneberger, Philipp Fischer et Thomas Brox. “U-net : Convolutional networks for biomedical image segmentation”. In : *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, p. 234-241.

[She+18] Yiru Shen et al. “Mining point cloud local structures by kernel correlation and graph pooling”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. T. 4. 2018.

References v

[SSP+03] Patrice Y Simard, David Steinkraus, John C Platt et al. “Best practices for convolutional neural networks applied to visual document analysis.”. In : *ICDAR*. T. 3. 2003, p. 958-962.

[Su+15] Hang Su et al. “Multi-view convolutional neural networks for 3D shape recognition”. In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 945-953.

[Su+18] Hang Su et al. “SPLATNet : Sparse Lattice Networks for Point Cloud Processing”. In : *arXiv preprint arXiv :1802.08275* (2018).

[Wan+18a] Shenlong Wang et al. “Deep parametric continuous convolutional neural networks”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 2589-2597.

[Wan+18b] Weiyue Wang et al. “SGPN : Similarity group proposal network for 3D point cloud instance segmentation”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, p. 2569-2578.

[Wan+18c] Yue Wang et al. “Dynamic graph CNN for learning on point clouds”. In : *arXiv preprint arXiv :1801.07829* (2018).

[WP15] Dominic Zeng Wang et Ingmar Posner. “Voting for Voting in Online Point Cloud Object Detection.”. In : *Robotics : Science and Systems*. T. 1. 2015, p. 5.

[WSS18] Chu Wang, Babak Samari et Kaleem Siddiqi. “Local Spectral Graph Convolution for Point Set Feature Learning”. In : *arXiv preprint arXiv :1803.05827* (2018).

References vi

[Wu+15] Zhirong Wu et al. "3D shapenets : A deep representation for volumetric shapes". In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, p. 1912-1920.

[Xu+18] Yifan Xu et al. "SpiderCNN : Deep Learning on Point Sets with Parameterized Convolutional Filters". In : *arXiv preprint arXiv :1803.11527* (2018).

[Yi+16] Li Yi et al. "A scalable active framework for region annotation in 3D shape collections". In : *ACM Transactions on Graphics (TOG)* 35.6 (2016), p. 210.

[Yi+17] Li Yi et al. "SyncSpecCNN : Synchronized spectral CNN for 3D shape segmentation". In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, p. 2282-2290.

[Arm+17] I. Armeni et al. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding". In : *ArXiv e-prints* (fév. 2017). arXiv : 1702.01105 [cs.CV].