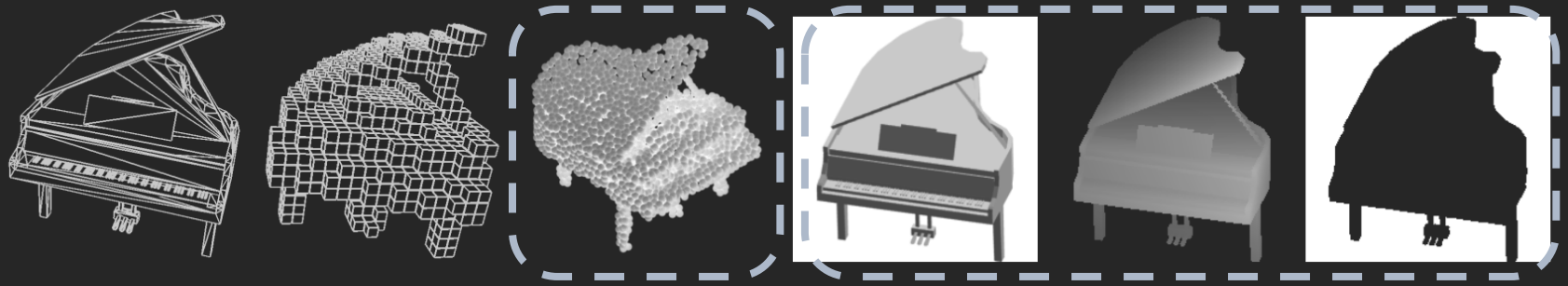
基于多视角图像与点云的三维特征学习

苏航 UMassAmherst

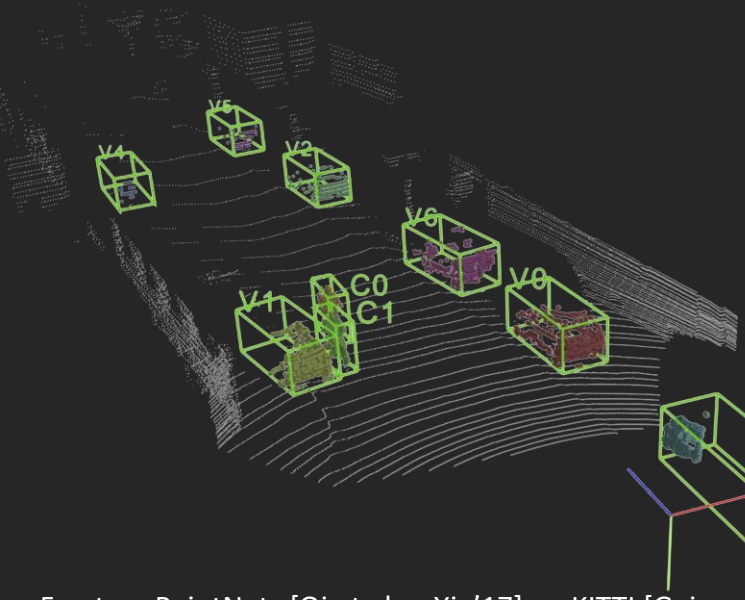
SPLATNet: Sparse Lattice Networks for Point Cloud Processing,

Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz, CVPR '18.

Representing 3D Data

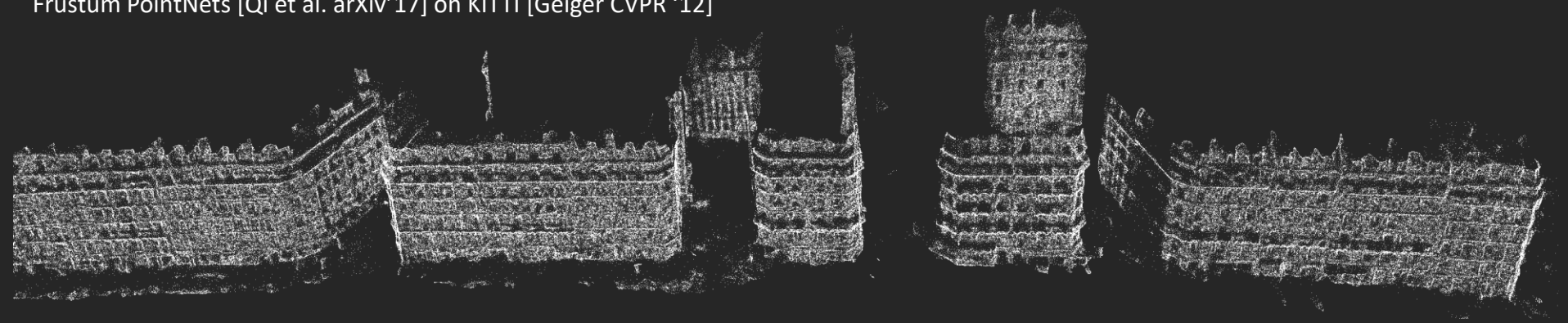


mesh voxels pointcloud 2D projections

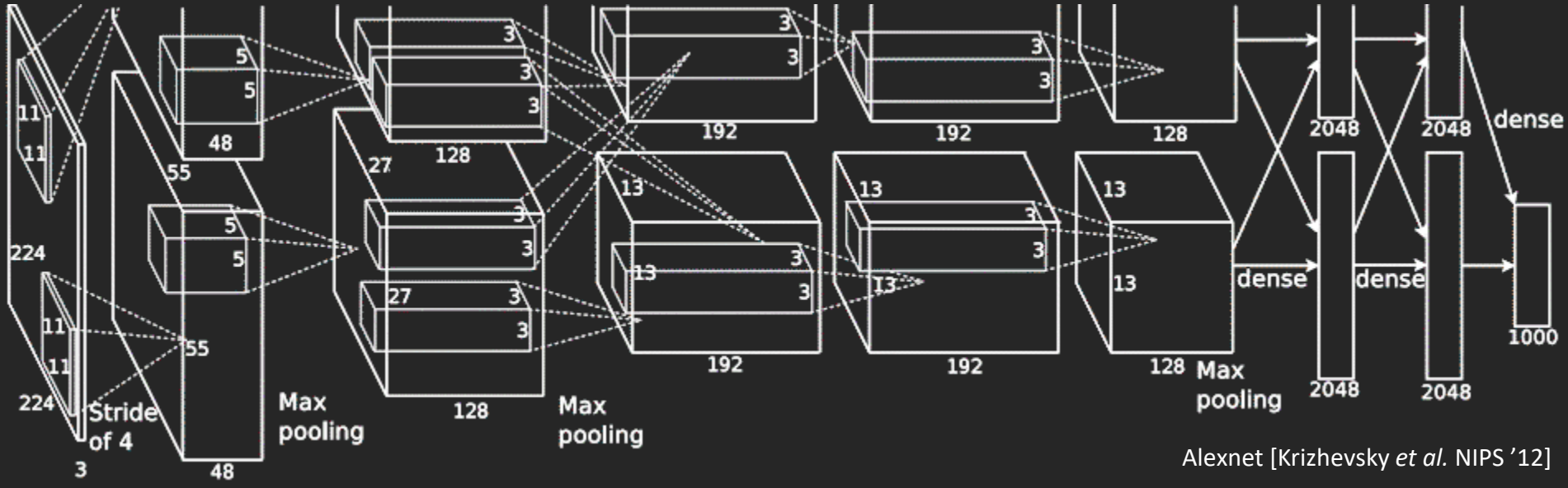


Frustum PointNets [Qi et al. arXiv'17] on KITTI [Geiger CVPR '12]

Cornell-RGBD [Koppula et al. NIPS '11]



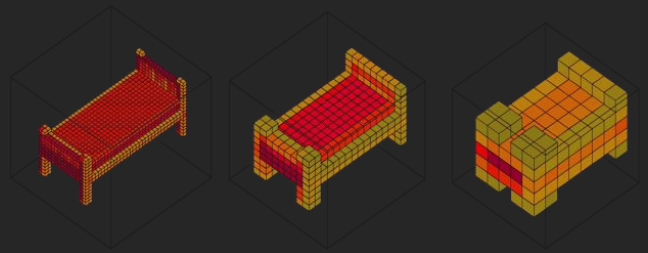
Ruemonge2014 [Riemenschneider et al. ECCV '14]



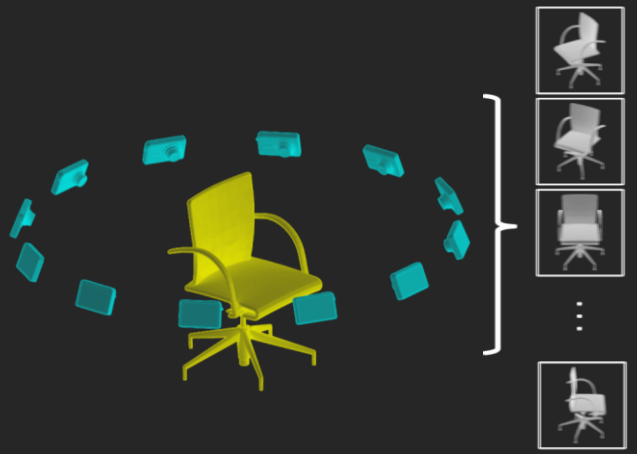
Alexnet [Krizhevsky *et al.* NIPS '12]

Not a good fit to standard CNNs

Workarounds



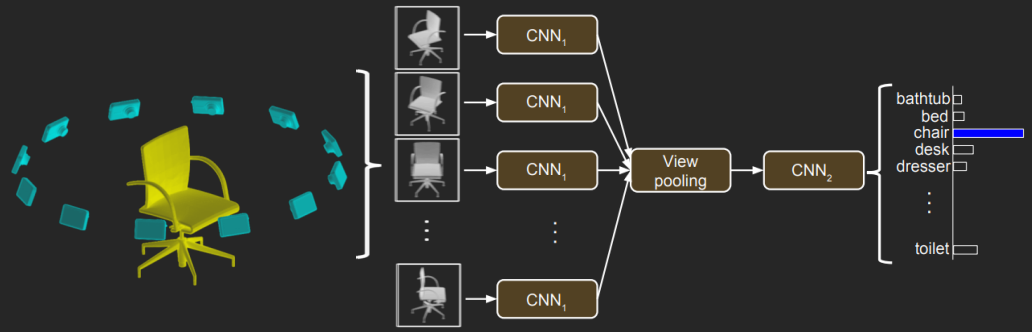
- Voxelization



- Multi-view 2D projections

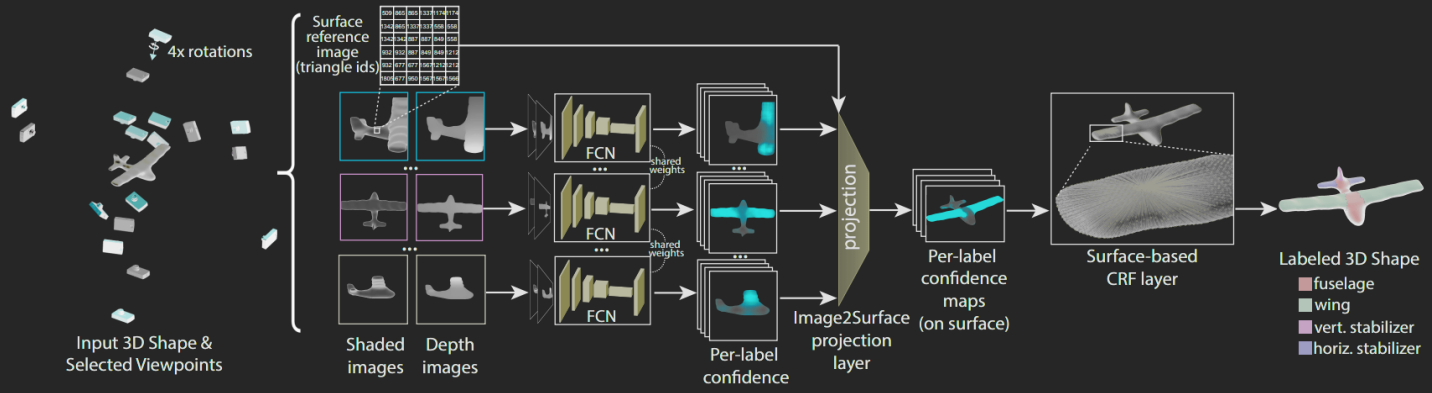
Figures from Octnet [Riegler *et al.* CVPR ‘17] and MVCNN [Su *et al.* ICCV ‘15]

Multi-view CNNs [Su et al. ICCV '15]

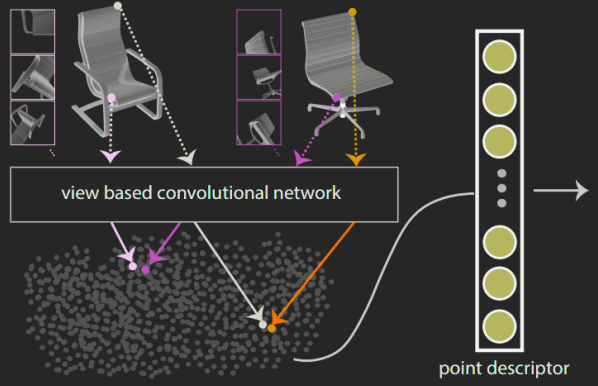


- Reduce the problem 3D → 2D
  1. Projection
  2. Extracting 2D features
  3. Feature fusion/pooling
  4. (optional) additional processing

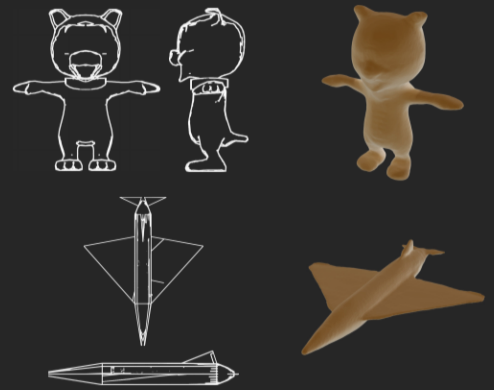
- Leverage strong 2D architectures and pre-trained models
- Computation efficiency



Mesh segmentation
E. Kalogerakis et al. CVPR '17



Local shape descriptor
H. Huang et al. Siggraph '18



Shape generation
Z. Lun 3DV '17

Multi-view CNNs

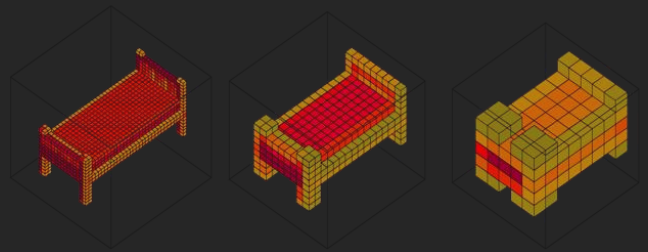
- Still a very competitive approach with some minor updates
  - Multi-Resolution [1]
  - Improved shading [2]
  - Stronger backbone [2]

	Input	Per Class Acc. (%)	Per Inst. Acc. (%)
Our MVCNN	Images	**92.4**	**95.0**
Rotation Net		-	94.8
Dominant Set Clustering		-	93.8
MVCNN-MultiRes		91.4	93.8
MVCNN		90.1	90.1
DynamicGraph	Point Clouds	90.2	92.2
Kd-Networks		-	91.8
PointNet++		-	90.7
PointNet		86.2	89.2
VRN Single	Voxels	-	91.3
O-CNN		-	90.6
VoxNet		-	83.0
3DShapeNets		77.3	84.7
PointNet++	PC+Normal	-	91.9
FusionNet	Voxels+Images	-	90.8

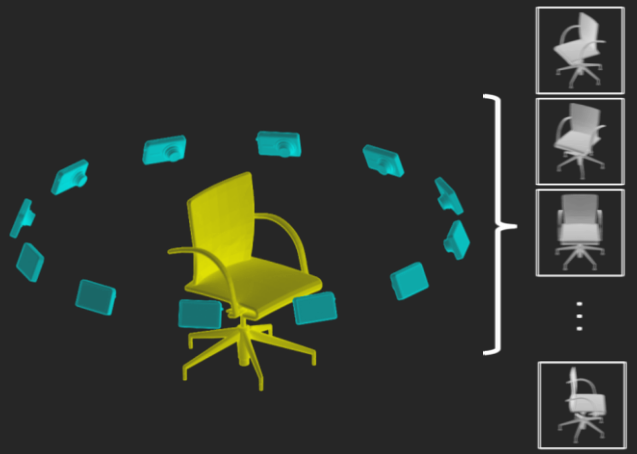
Table from [2]

[1] C. Qi et al. Volumetric and Multi-View CNNs for Object Classification on 3D Data CVPR '16.
[2] J. Su et al. A Deeper Look at 3D Shape Classifiers. arXiv:1809.02560.

Workarounds



- Voxelization

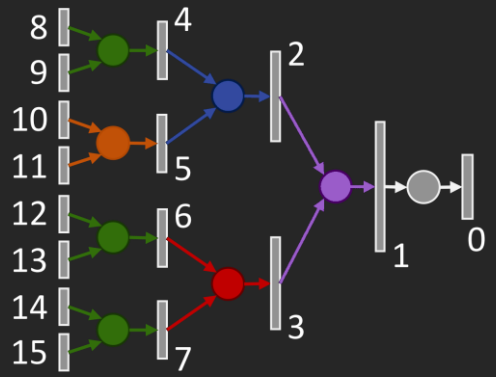


- Multi-view 2D projections

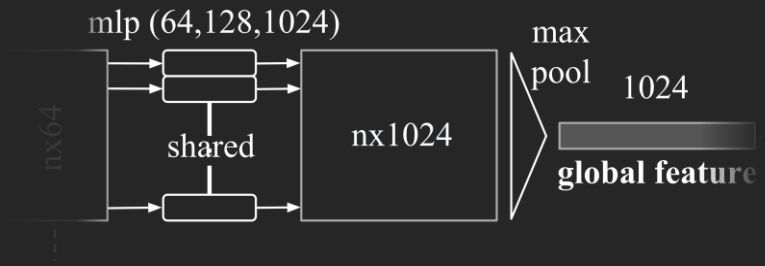
Figures from Octnet [Riegler *et al.* CVPR ‘17] and MVCNN [Su *et al.* ICCV ‘15]

Recent approaches

Kd-Networks
[Klokov *et al.* ICCV '17]



PointNet/PointNet++
[Qi *et al.* CVPR ’17 & NIPS ’17]

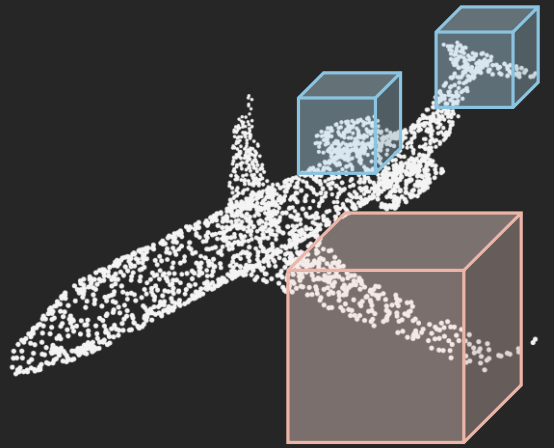


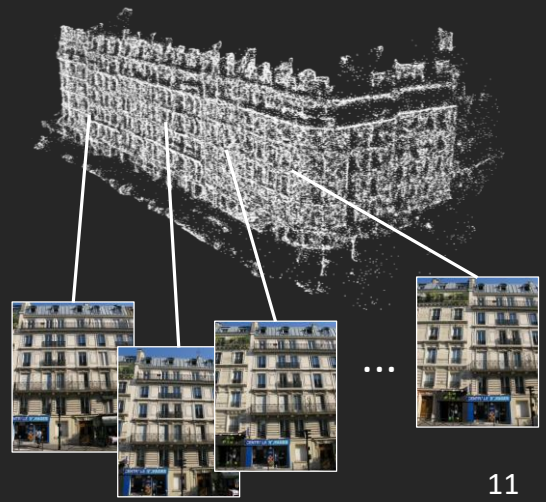
Goal

Operate on point clouds directly

Two key capabilities:

1. Flexible specification of receptive fields
2. Joint 2D-3D processing

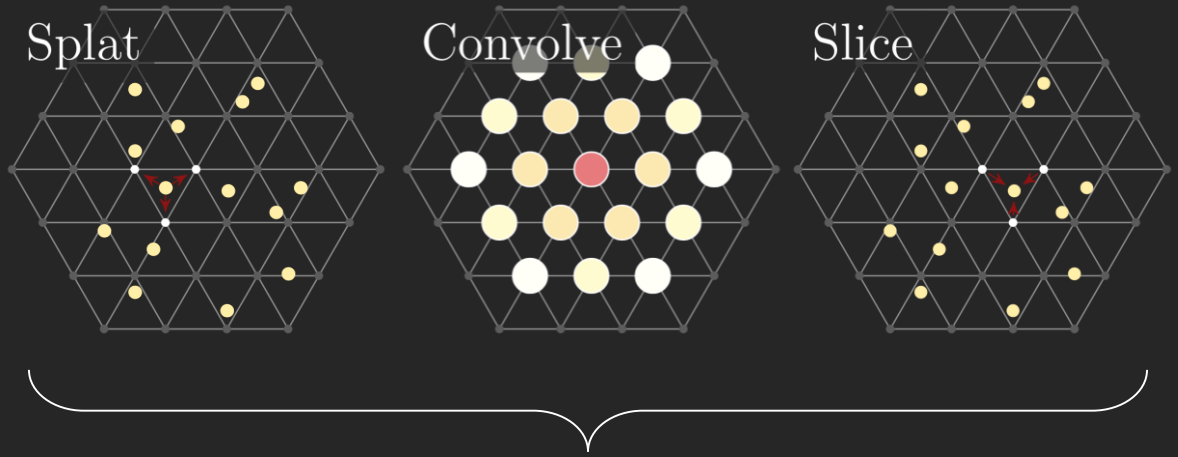




Bilateral Convolution Layer (BCL)

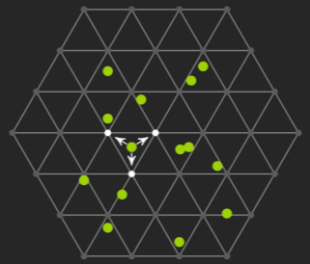
[Jampani *et al.* CVPR ’16] [Kiefel *et al.* ICLR ‘15 workshops]

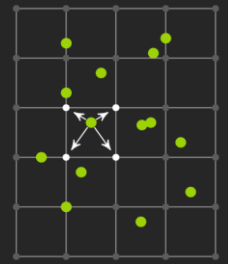
Convolution on sparse unordered points

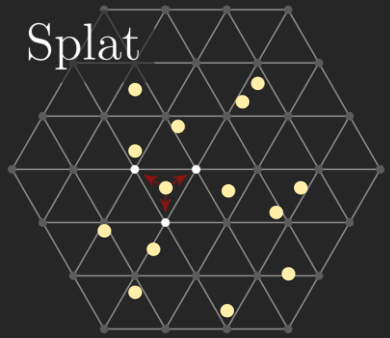


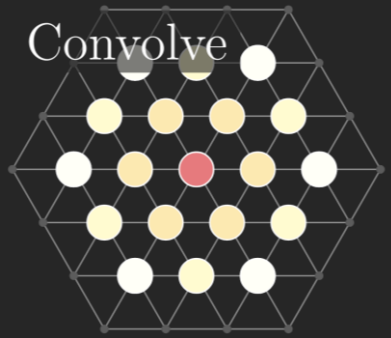
3 steps in BCL

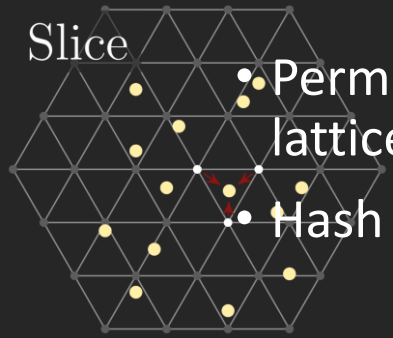
Efficient computation



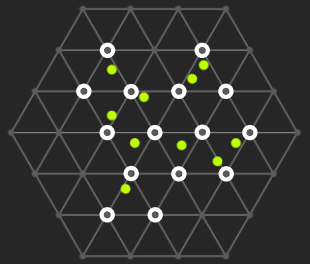








- Permutohedral lattice [1]
- Hash table



3 steps in BCL

[1] Adams *et al.* Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum ‘10

Point feature and lattice feature

"what" "where"

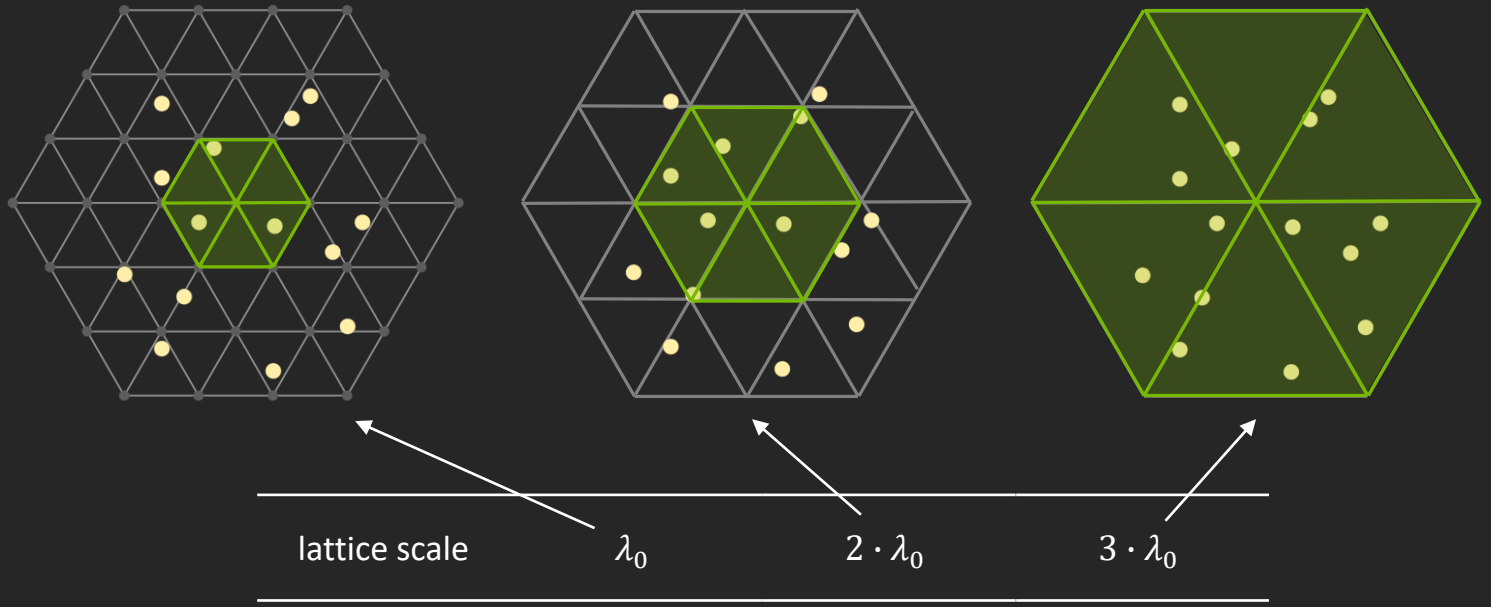
Image

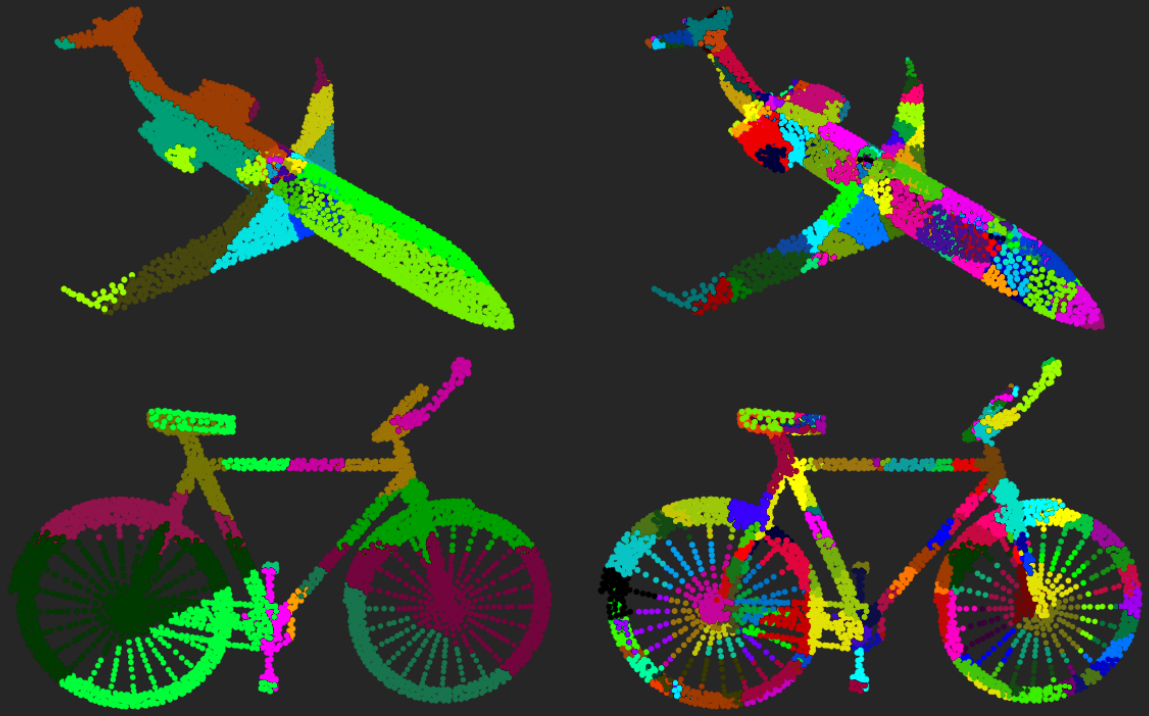
point feature	(r, g, b)	$f(\dots)$
lattice feature	(x, y)	(x, y)

Point cloud

point feature	1	(x, y, z)	(r, g, b)	$f(\dots)$
lattice feature	(x, y, z)	(x, y, z)	(x, y, z, n_x, n_y, n_z)	(x, y, z)

Controlling receptive fields





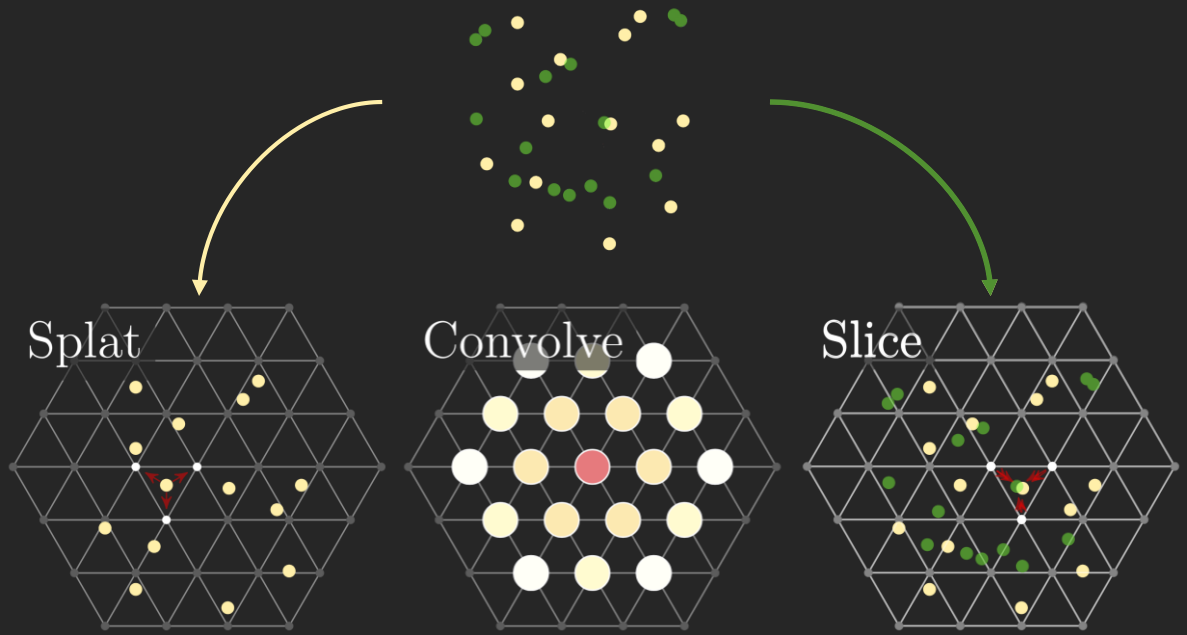
lattice feature: (x, y, z)

lattice scale: $8 \cdot \lambda_0$

lattice feature: (x, y, z)

lattice scale: λ_0

Input and output can be at different points



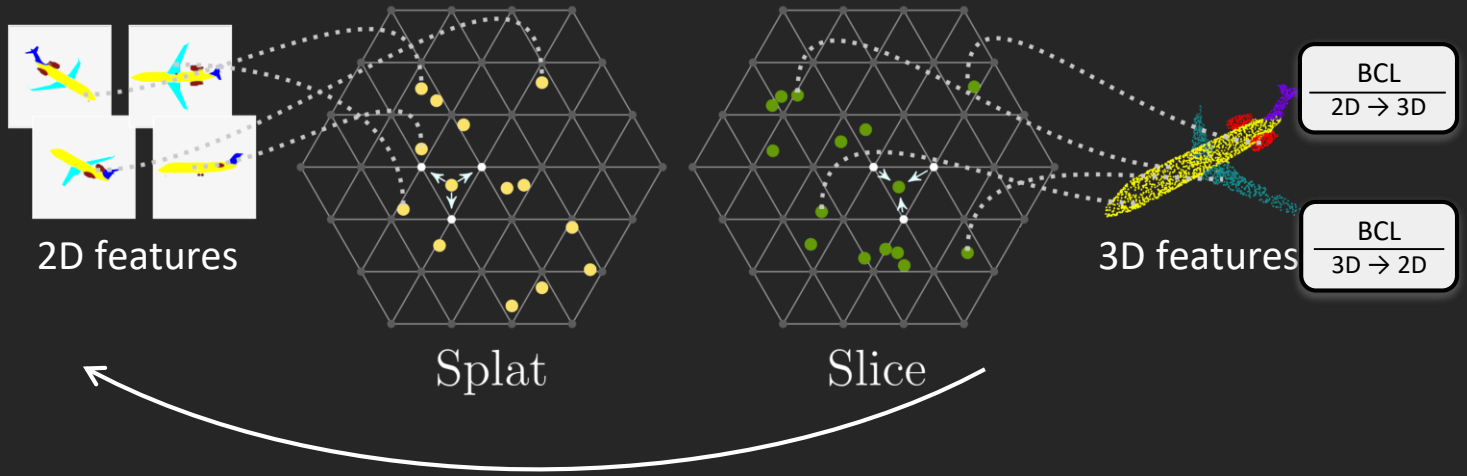
Smooth projection between 2D and 3D

- Each 2D pixel p: $f(p)$, (x, y, z)

2D features

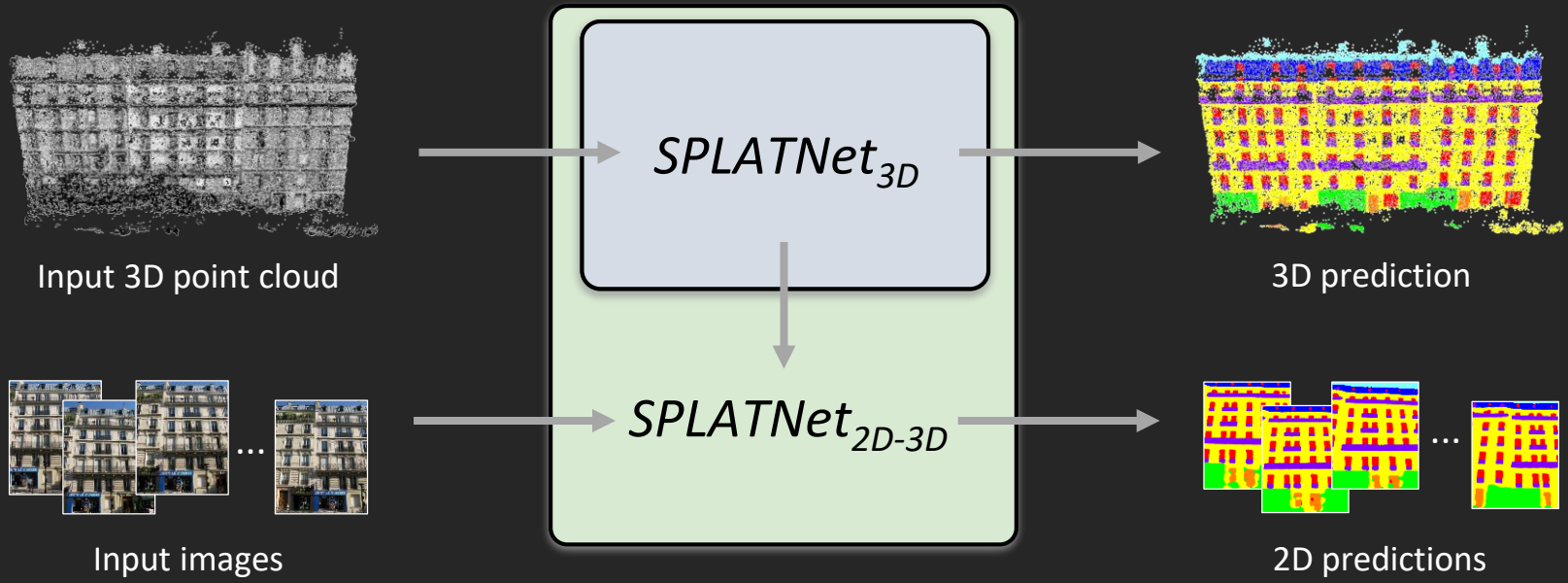
Smooth projection between 2D and 3D

- Each 2D pixel p: $f(p)$, (x, y, z)
  - → Point feature: $f(p)$
  - → Lattice feature: (x, y, z)

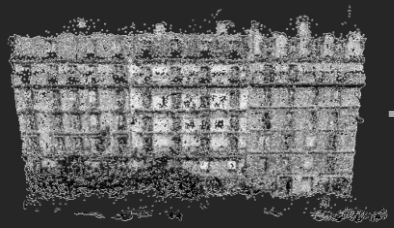


SPLATNet architecture

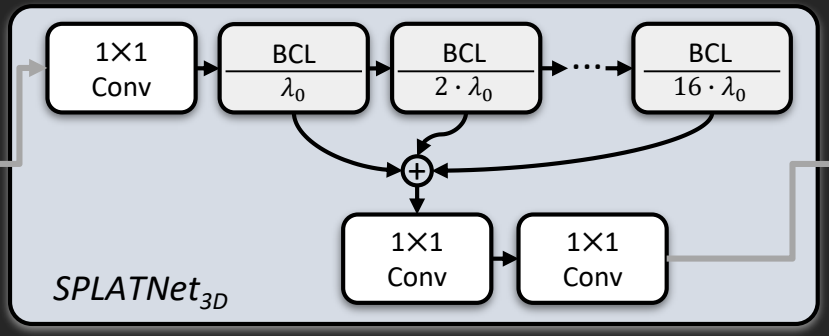
SPLATNet architecture

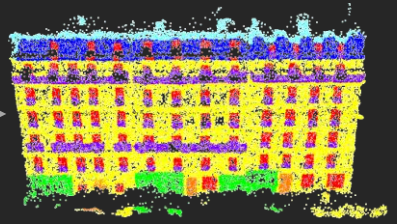


$SPLATNet_{3D}$



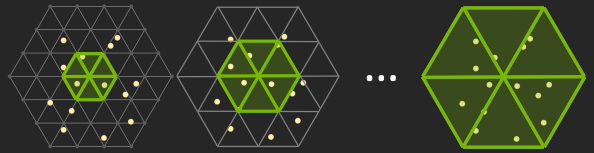
Input 3D point cloud

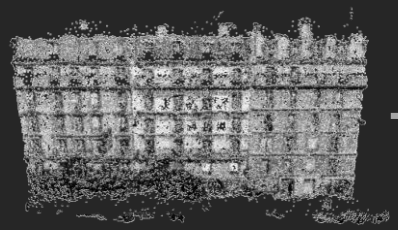




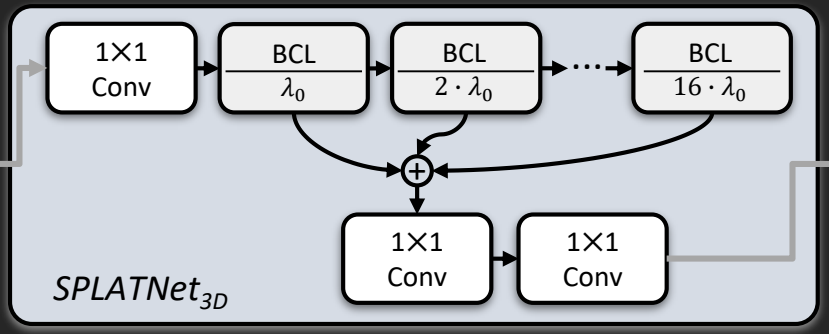
3D prediction

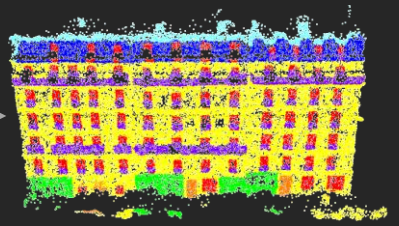
$SPLATNet_{3D}$





Input 3D point cloud

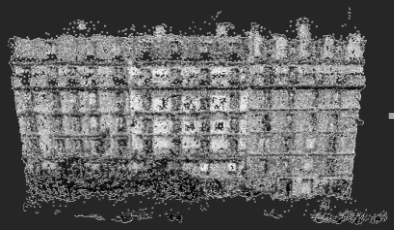




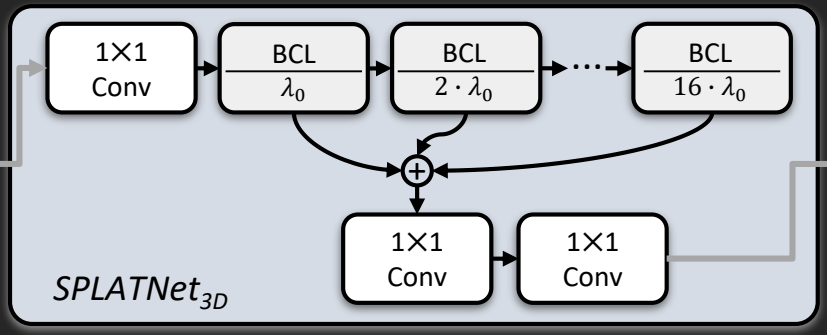
3D prediction

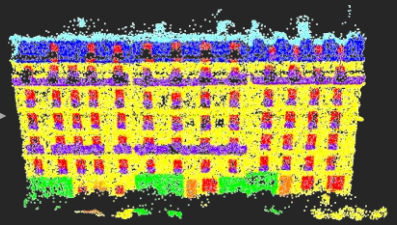
⊕ concatenation

$SPLATNet_{2D\text{-}3D}$



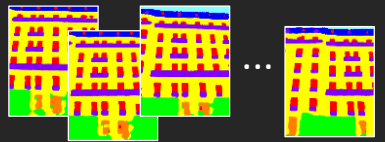
Input 3D point cloud





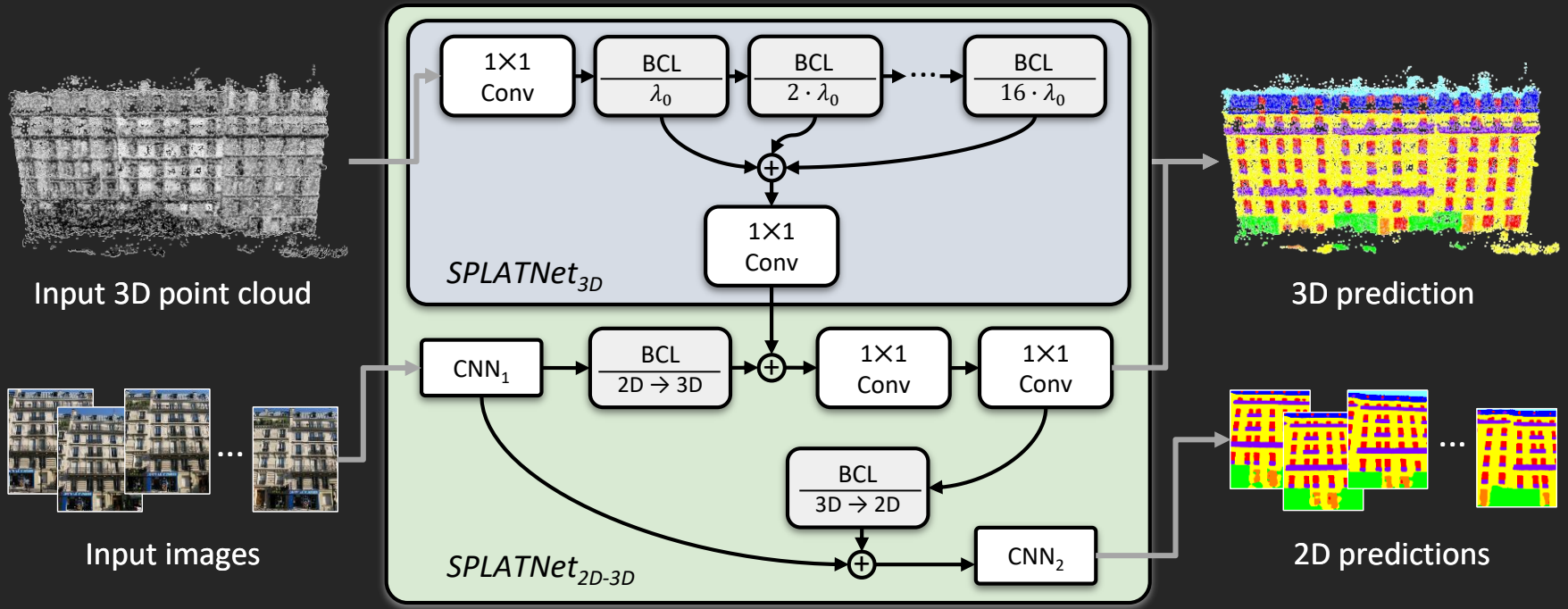
3D prediction

Input images

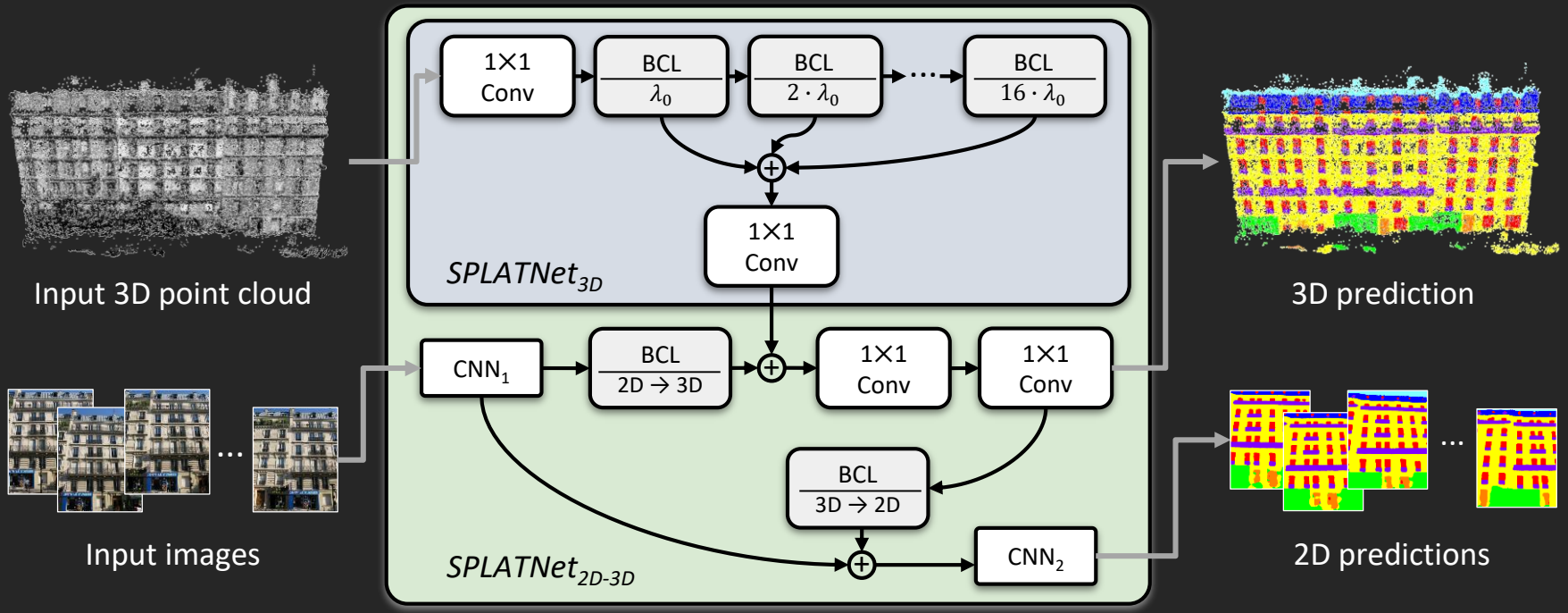


2D predictions

$SPLATNet_{2D\text{-}3D}$



$SPLATNet_{2D\text{-}3D}$



Experiments

Facade segmentation (Ruemonge2014 [1])

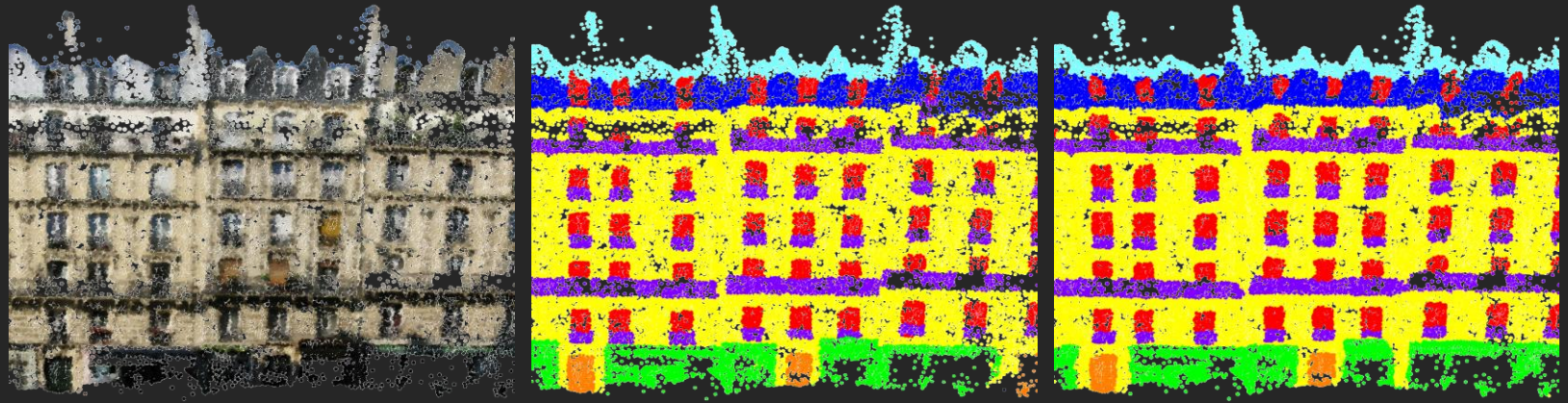
with only 3D data

	IoU	runtime (min)
OctNet [2]	59.2	-
Autocontext$_{3D}$ [3]	54.4	16
SPLATNet$_{3D}$	**65.4**	**0.06**

with both 2D & 3D data

	IoU	runtime (min)
Autocontext$_{2D-3D}$ [3]	62.9	87
SPLATNet$_{2D-3D}$	**69.8**	**1.2**

[2] Riegler *et al.* CVPR '17 [3] Gadde *et al.* PAMI '17



Input point cloud | Ground-truth | Ours (*SPLATNET$_{2D-3D}$*)

[1] Riemenschneider et al. ECCV '14

2D predictions

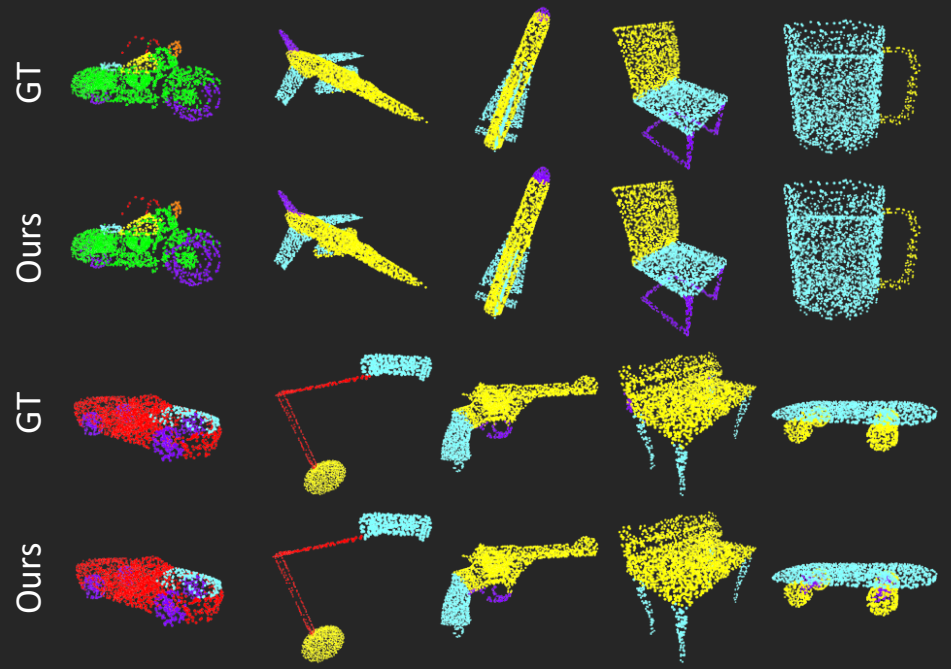
	IoU	runtime (min)
Autocontext$_{2D}$ [1]	60.5	117
Autocontext$_{2D-3D}$ [1]	62.7	146
2D CNN [2] only	69.3	0.84
SPLATNet$_{2D-3D}$	70.6	4.34

[1] Gadde *et al.* PAMI '17
[2] Chen *et al.* ICLR '15

Input image | Ground-truth | Our prediction

3D object part labeling (ShapeNet [1])

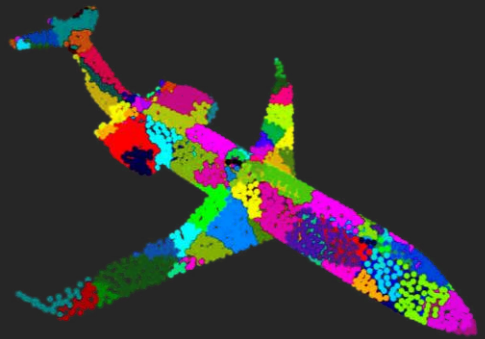
	class avg. IoU	instance avg. IoU
Yi *et al.* [1]	79.0	81.4
3DCNN [2]	74.9	79.4
Kd-network [3]	77.4	82.3
PointNet [2]	80.4	83.7
PointNet++ [4]	81.9	85.1
SyncSpecCNN [5]	82.0	84.7
SPLATNet_{3D}	82.0	84.6
SPLATNet_{2D-3D}	83.7	85.4



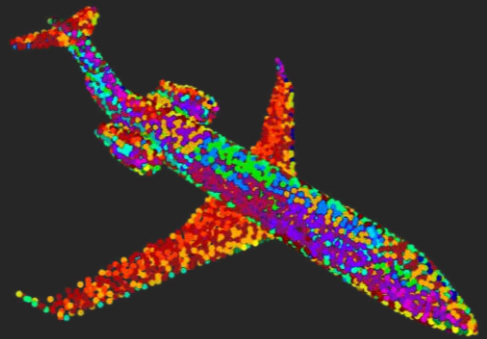
[1] Yi *et al.* SIGGGRAPH Asia '16
[2] Qi *et al.* CVPR '17
[3] Klokov and Lempitsky ICCV '17
[4] Qi *et al.* NIPS '17
[5] Yi *et al.* CVPR '17

Filtering in other lattice spaces

- Adding additional (x, y, z, n_x, n_y, n_z) filtering → +0.2 IoU

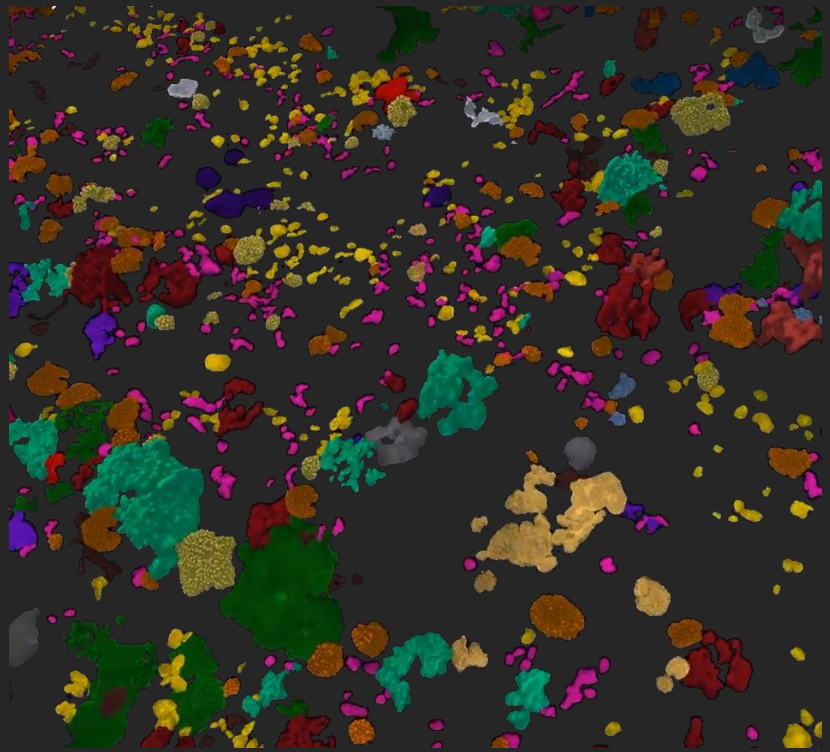


(x, y, z)



(n_x, n_y, n_z)

Real-world application on corals!

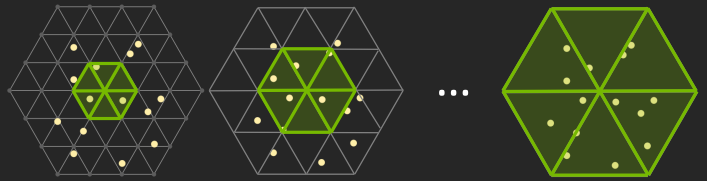


http://100islandchallenge.org/

SPLATNet: Summary

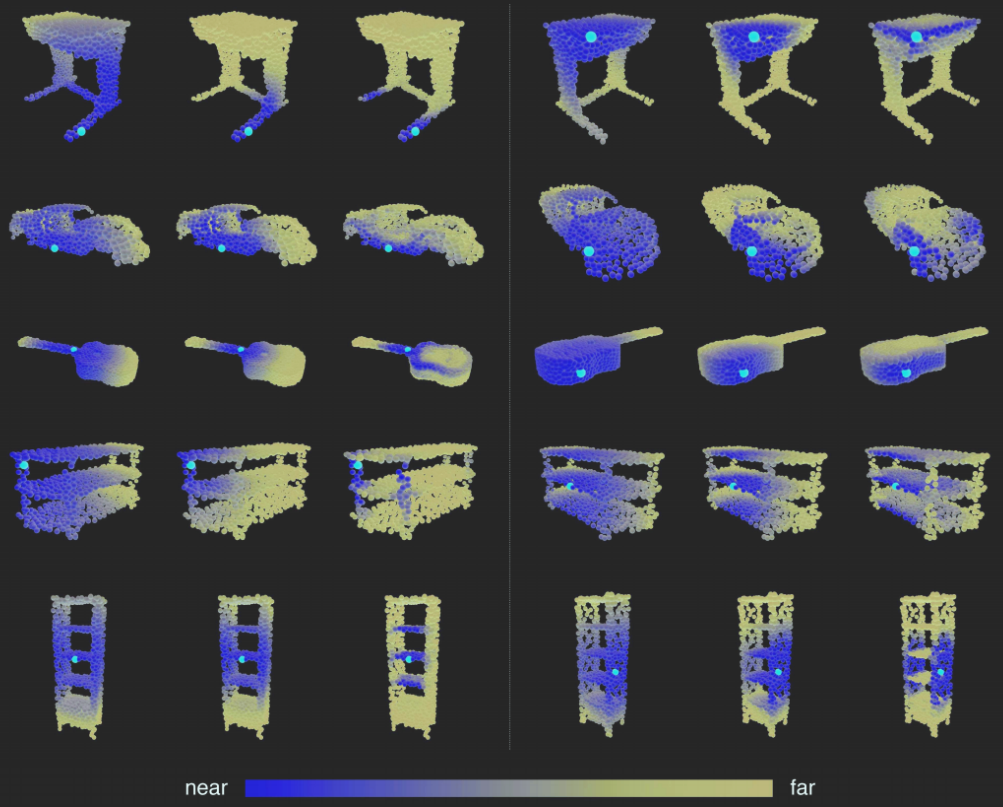
- efficient computation directly on point clouds
- flexible specifications of receptive fields
- seamless joint 2D-3D processing

https://github.com/nvlabs/*SPLATNet*



Follow-up work

- Learning lattice feature
- Dense version with more efficient implementation



Y. Wang et al. Dynamic Graph CNN for Learning on Point Clouds. arXiv:1801.07829