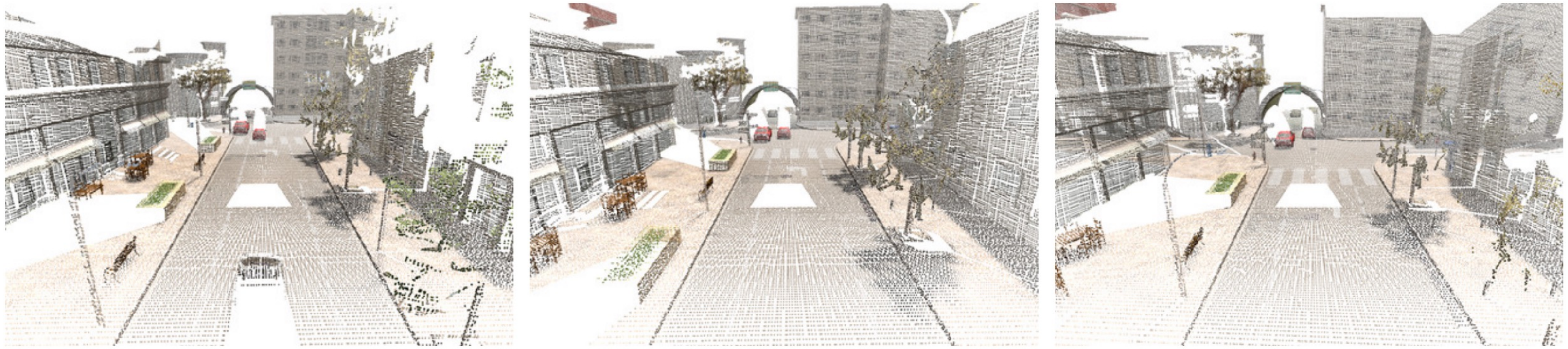
4D Spatio-Temporal ConvNets

Minkowski Convolutional Neural Networks

Introduction

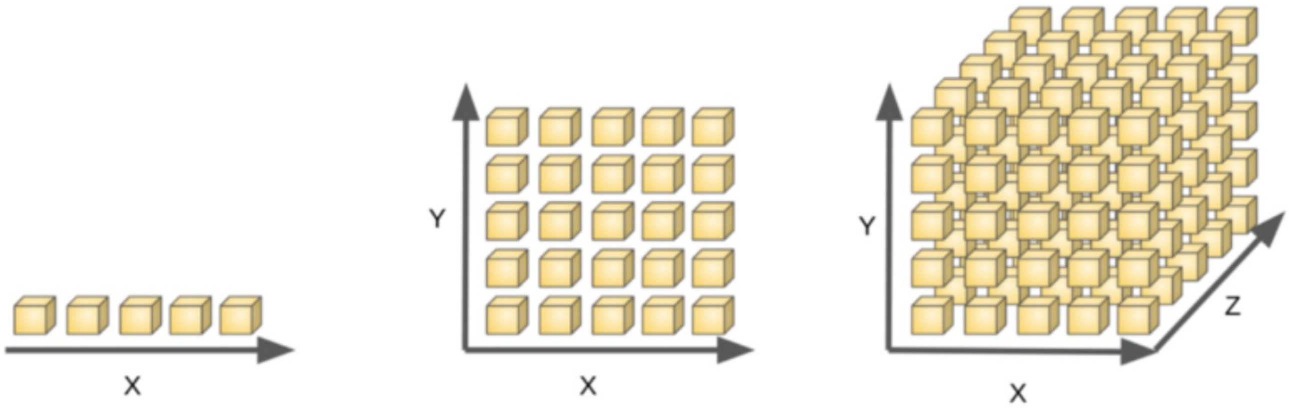
- 4D Space: 3 spatial dim., 1 time dim.
- 3D Pointcloud as a slice along the time axis
- Captured by depth sensing sensors over time
- Problem statement:
  - Input: 4D data
  - Goal: Segmentation of points



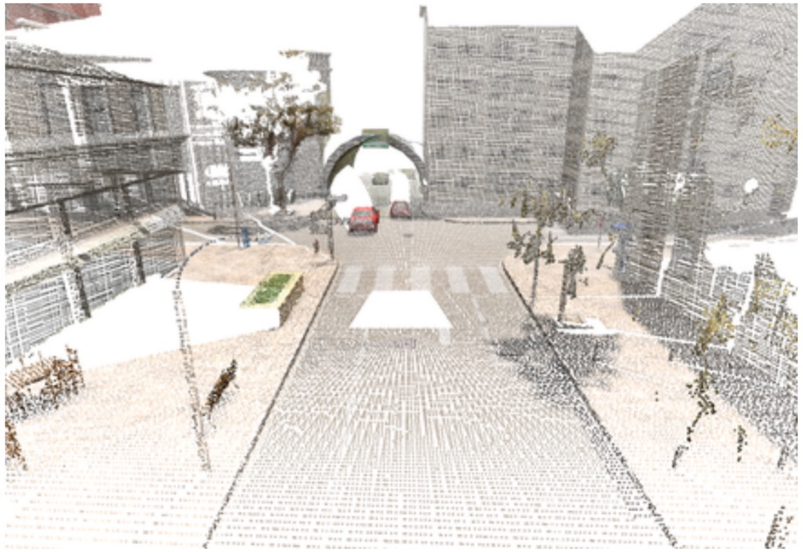
Source: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Motivation

- More and more 4D data, e.g. in robotics
- But: Curse of dimensionality
  - Exponential computational effort
  - Sparse data

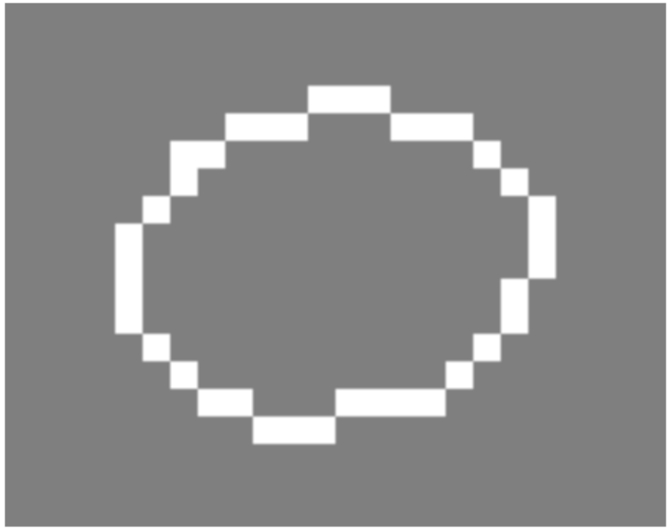


Source: https://www.freecodecamp.org/news/the-curse-of-dimensionality-how-we-can-save-big-data-from-itself-d9fa0f872335/



Source: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Concepts explained – Sparse Data



Source: https://github.com/facebookresearch/SparseConvNet

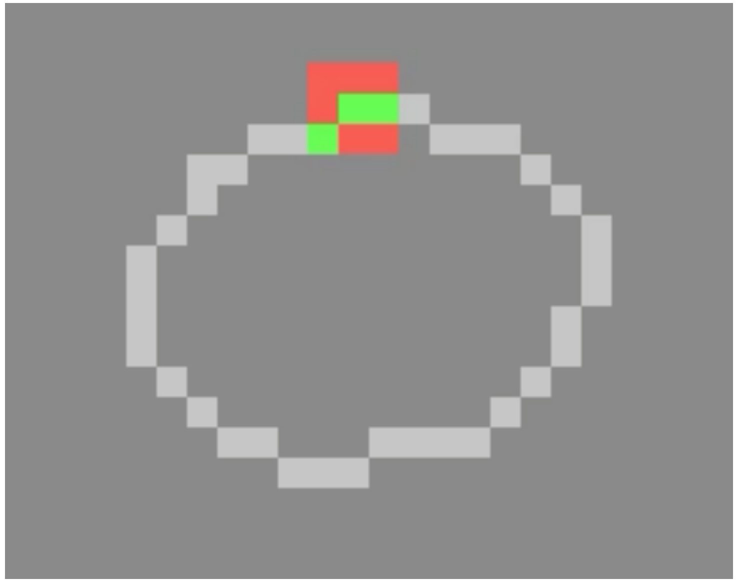
[Coordinate],
[Value]

From: $28 * 28 = 784\ values$

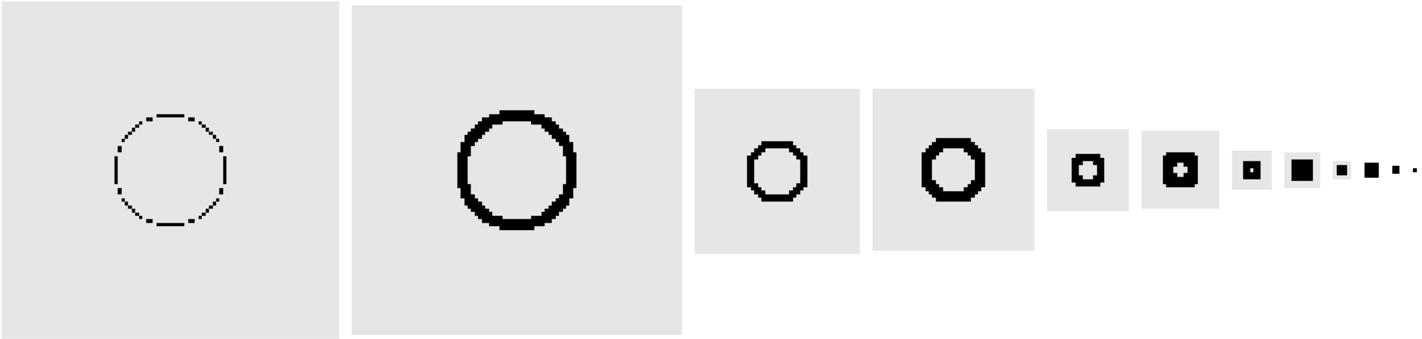
To: $50 * 2 + 50 * 1 = 150\ values$

Concepts explained – Sparse Convolutions

- Non-trivial convolutions – mapping from input to output
- Computation for just some outputs

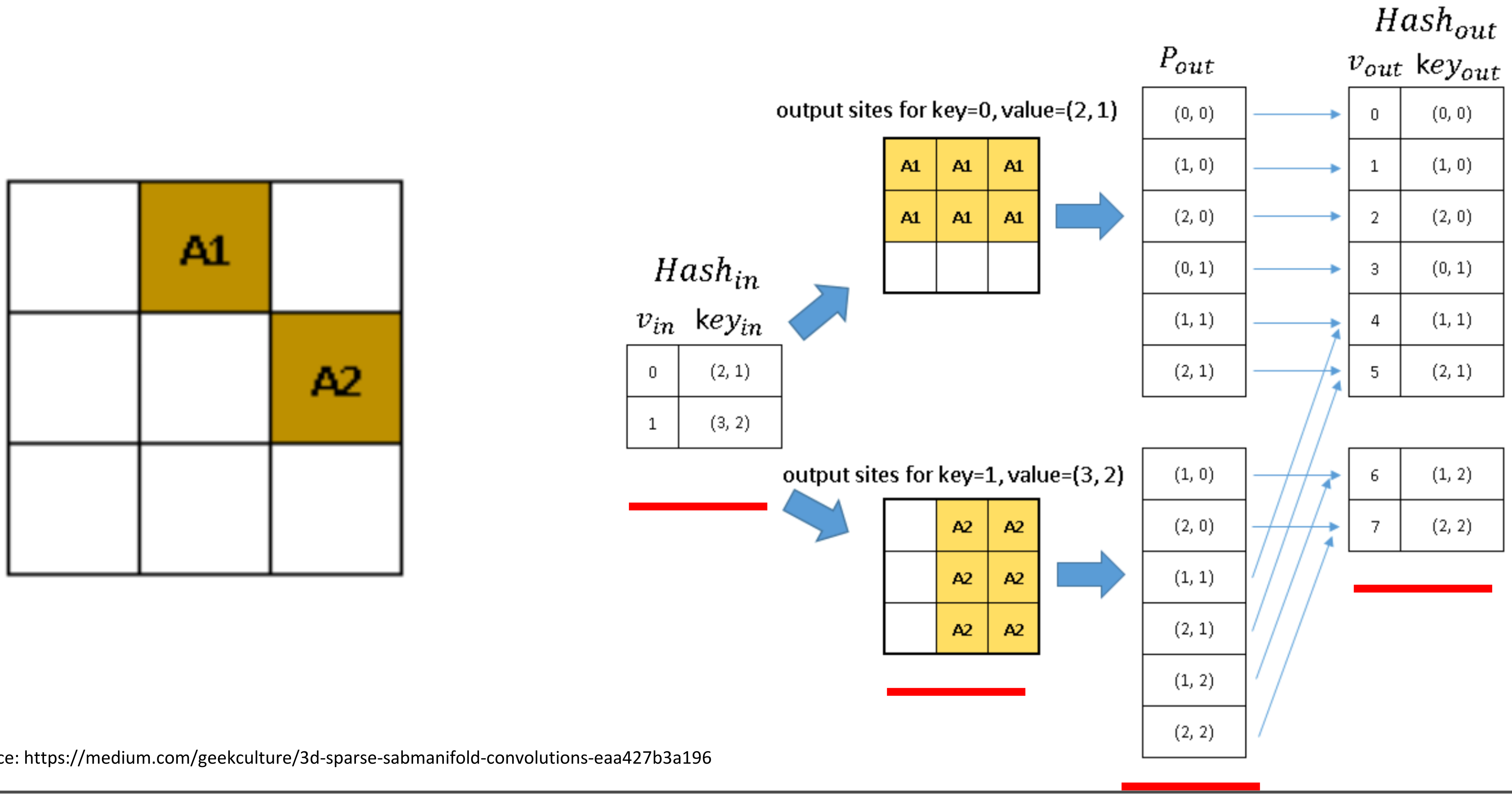


Source: https://github.com/facebookresearch/SparseConvNet



Source: Spatially-sparse convolutional neural networks

Concepts explained – Sparse Convolutions



Source: https://medium.com/geekculture/3d-sparse-sabmanifold-convolutions-eaa427b3a196

Concepts explained – Sparsity

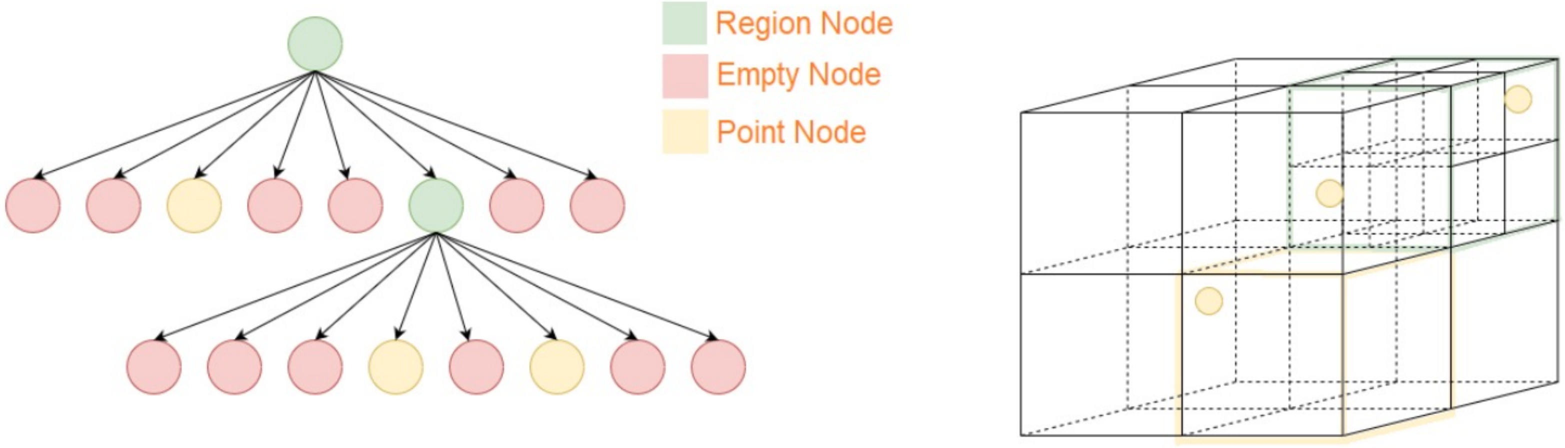
- Deals with the curse of dimensionality
- Sparse representation – more memory efficient
- Sparse convolutions – less computational complexity

Related Work

- 4D perception without NN
  - 4D Markov random field
  - Spatio-temporal CNN
- 2D video perception
- Related Work is in 3D Perception
  - Perception utilizing 3D convolutions
  - Perception without 3D convolutions

Related Work – 3D Convolutions

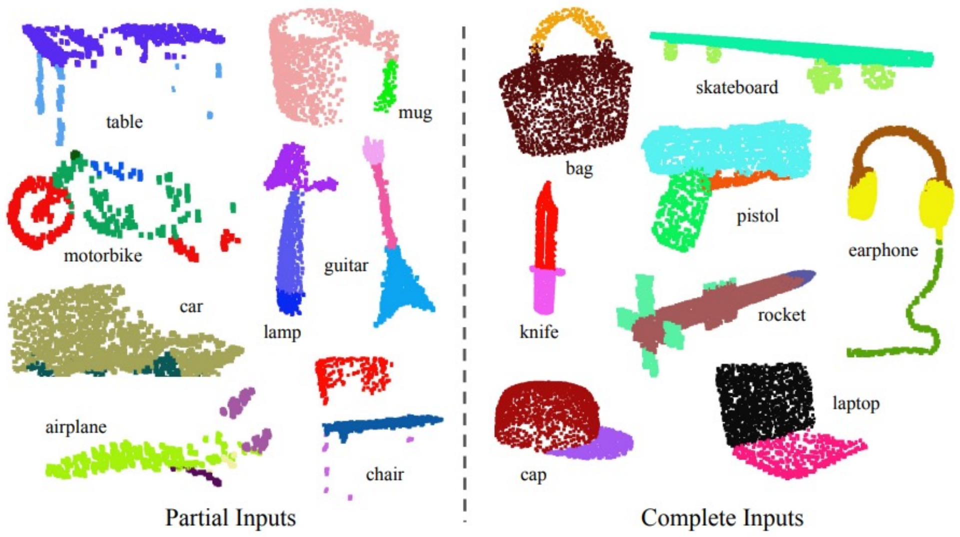
- Dense 3D convolutions
  - Suffer from the curse of dim. → OctNet
- Sparse 3D convolutions
- Continuous kernels in a continuous space



Source: https://iq.opengenus.org/octree/

Related Work –Without 3D Convolutions

- 2D convolutions for 3D perception
- Perception on pointclouds
  - PointNet/ PointNet++



PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

Methods – Sparse Tensors

- Coordinates with respective values
- 3 spatial dimensions and 1 temporal dimension
- Data needs to be quantized

$$C = \begin{bmatrix} x_1 & y_1 & z_1 & t_1 & b_1 \\ & & \vdots & & \\ x_N & y_N & z_N & t_N & b_N \end{bmatrix}, \; F = \begin{bmatrix} \mathbf{f}_1^T \\ \vdots \\ \mathbf{f}_N^T \end{bmatrix}$$

Methods – Generalized (sparse) Convolutions

1. $$\mathbf{x}^{\text{out}}_{\mathbf{u}} = \sum_{\boxed{\mathbf{i} \in \mathcal{V}^D(K)}} W_{\mathbf{i}} \mathbf{x}^{\text{in}}_{\mathbf{u}+\mathbf{i}} \; \boxed{\text{for } \mathbf{u} \in \mathbb{Z}^D}$$

2. $$\mathbf{x}^{\text{out}}_{\mathbf{u}} = \sum_{\boxed{\mathbf{i} \in \mathcal{N}^D(\mathbf{u}, \mathcal{C}^{\text{in}})}} W_{\mathbf{i}} \mathbf{x}^{\text{in}}_{\mathbf{u}+\mathbf{i}} \; \boxed{\text{for } \mathbf{u} \in \mathcal{C}^{\text{out}}}$$

➢ We need an algorithm for this

Methods – Pooling Operations

- Max Pooling
- Global Pooling
- Average Pooling
- Sum Pooling

$\Bigg\}$ Non-trivial algorithm

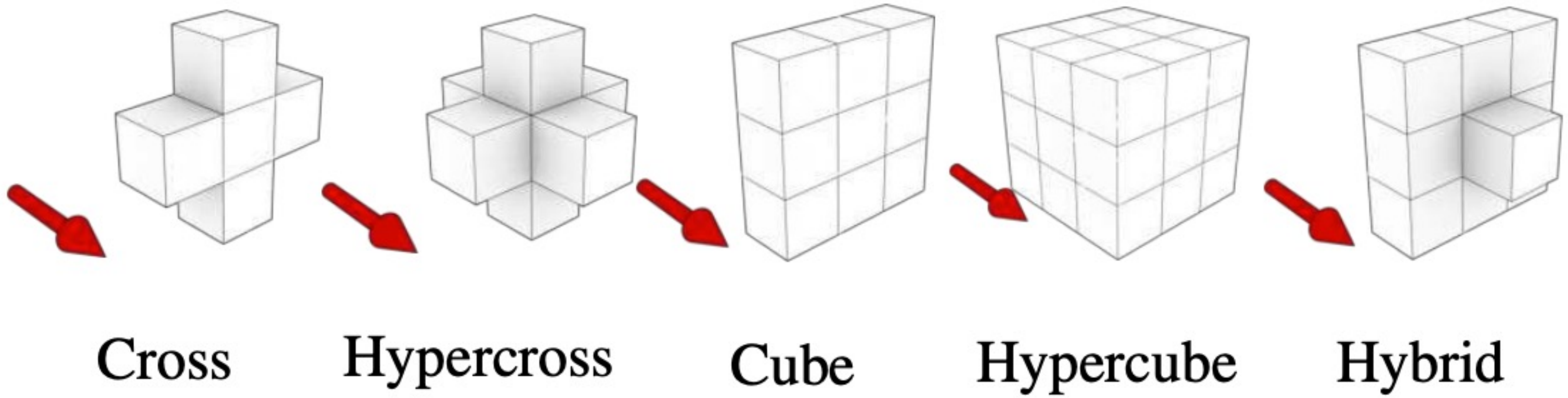
- Non-linearity

$\}$ Trivial

Methods - Kernel Shapes

- Surfaces of objects in 3D are to be perceived
- Kernels are overparametrized

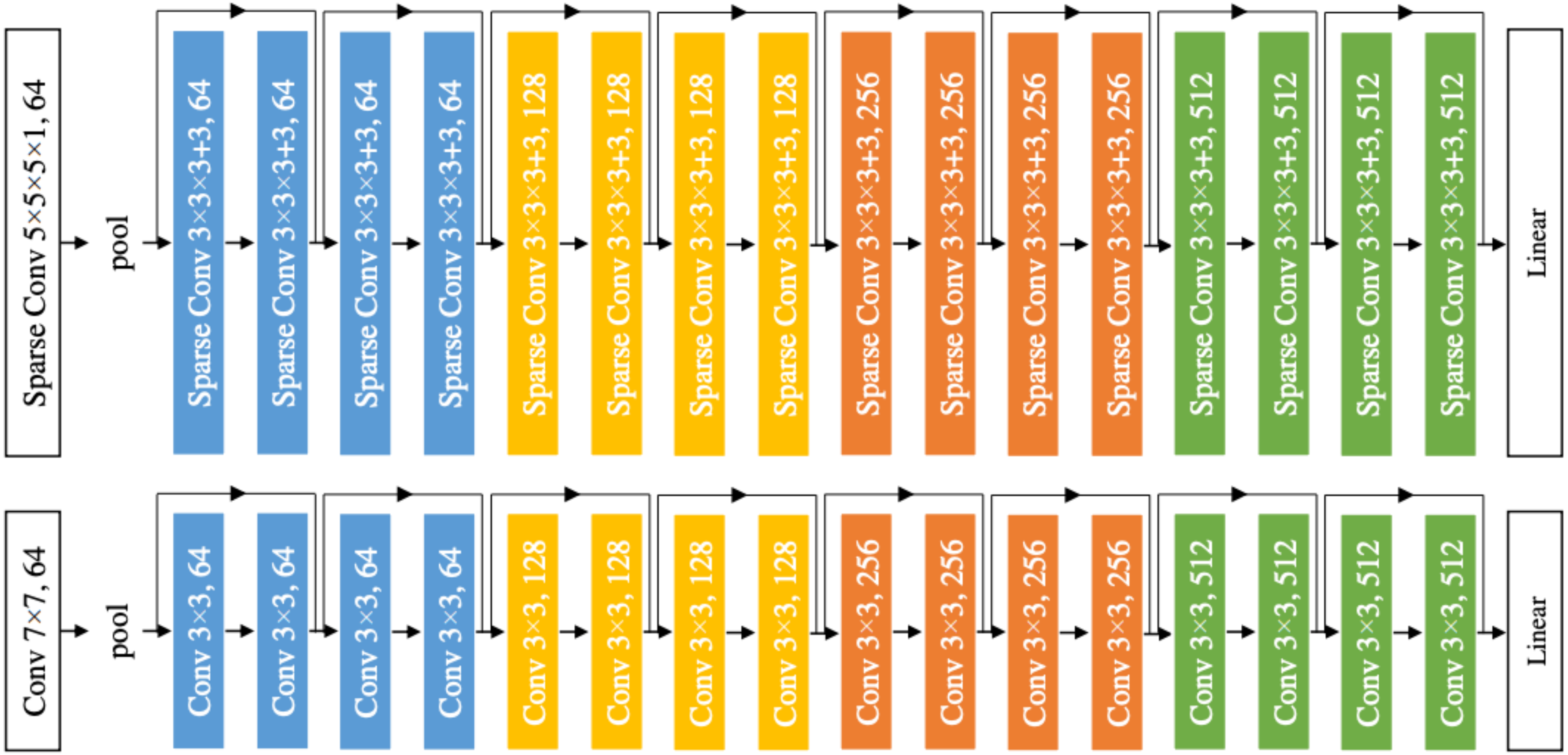
➢ Reduce params in kernels



Source: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

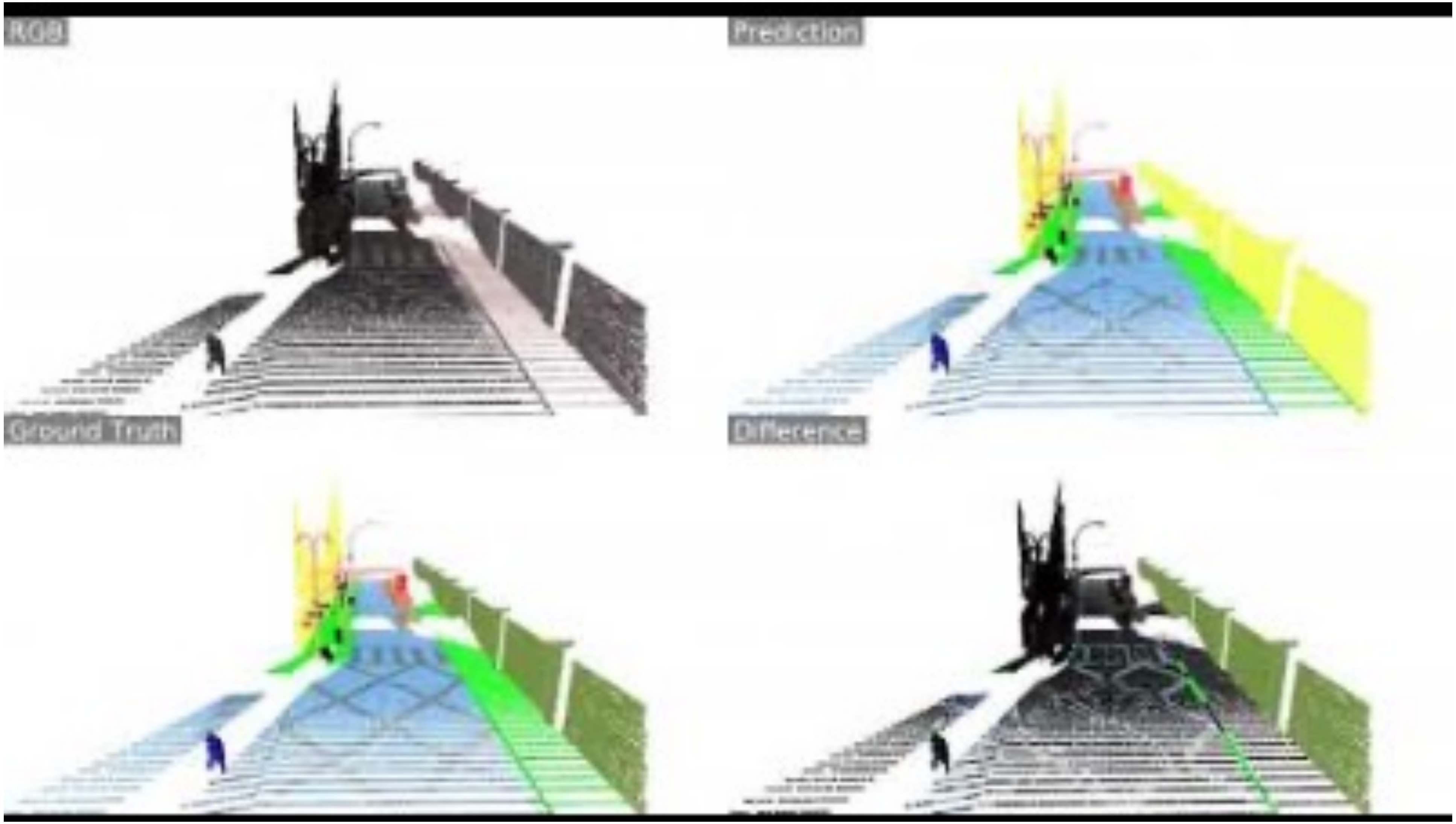
Methods - Full Architecture

4D input



2D input

Results - Video



Results – ScanNet

Source: http://www.scan-net.org/

Table 1: 3D Semantic Label Benchmark on ScanNet† [5]

Method	mIOU
ScanNet [5]	30.6
SSC-UNet [10]	30.8
PointNet++ [24]	33.9
ScanNet-FTSDF	38.3
SPLATNet [29]	39.3
TangetConv [30]	43.8
SurfaceConv [21]	44.2
3DMV‡ [6]	48.4
3DMV-FTSDF‡	50.1
PointNet++SW	52.3
MinkowskiNet42 (5cm)	**67.9**
MinkowskiNet42 (2cm)†	72.1
SparseConvNet [10]†	**72.5**

†: post-CVPR submissions. ‡: uses 2D images additionally. Per class IoU in the supplementary material. The parenthesis next to our methods indicate the voxel size.

Source: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks

Results – Synthia 4D

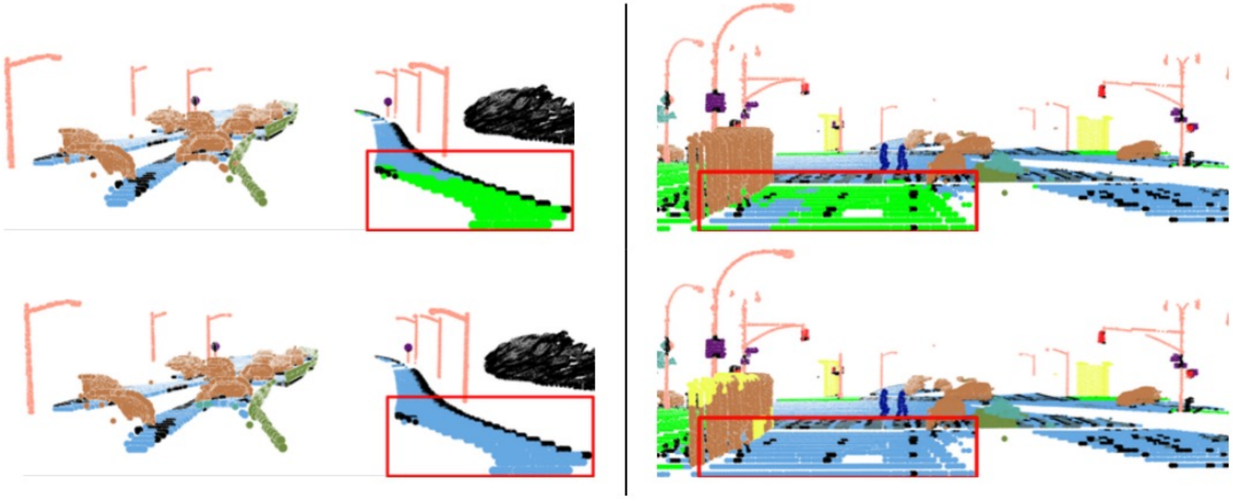
Table 3: Segmentation results on the noisy Synthia 4D dataset

IoU	Building	Road	Sidewalk	Fence	Vegetation	Pole	Car	Traffic Sign	Pedestrian	Lanemarking	Traffic Light	mIoU
3D MinkNet42	87.954	97.511	78.346	84.307	96.225	94.785	87.370	42.705	**66.666**	52.665	55.353	76.717
3D MinkNet42 + TA	87.796	97.068	78.500	83.938	96.290	94.764	85.248	43.723	62.048	50.319	54.825	75.865
4D Tesseract MinkNet42	**89.957**	96.917	81.755	82.841	96.556	96.042	91.196	52.149	51.824	**70.388**	**57.960**	78.871
4D MinkNet42	88.890	**97.720**	**85.206**	**84.855**	**97.325**	**96.147**	**92.209**	**61.794**	61.647	55.673	56.735	**79.836**

TA denotes temporal averaging. As the input pointcloud coordinates are noisy, averaging along the temporal dimension introduces noise.

3D:

4D:



Conclusion – Main Contribution/ Key-Idea

- Generalizes (sparse) convolutions to any dimension
- Introduces different kernel shapes

➢ Coming up with concrete algorithms for these ideas

Conclusion – Limitations

- Deep Learning on Dynamic 3D Point Cloud Sequences:
  - “Quantization introduces error”
  - Their method looks at the neighbours of each point instead of the space itself

Q&A