Learning Joint 2D-3D Representations for Depth Completion

ICCV 2019

2020.02

Sungha Choi

Contents

Deep Parametric Continuous Convolutional Neural Networks

Shenlong Wang[1,3,*] Simon Suo[2,3,*] Wei-Chiu Ma[3] Andrei Pokrovsky[3] Raquel Urtasun[1,3]
[1]University of Toronto, [2]University of Waterloo, [3]Uber Advanced Technologies Group
{slwang, suo, weichiu, andrei, urtasun}@uber.com

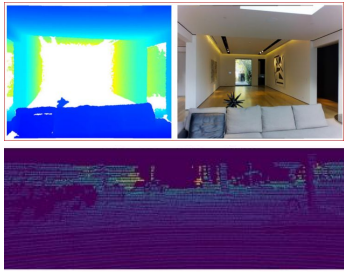
CVPR 2018

Learning Joint 2D-3D Representations for Depth Completion

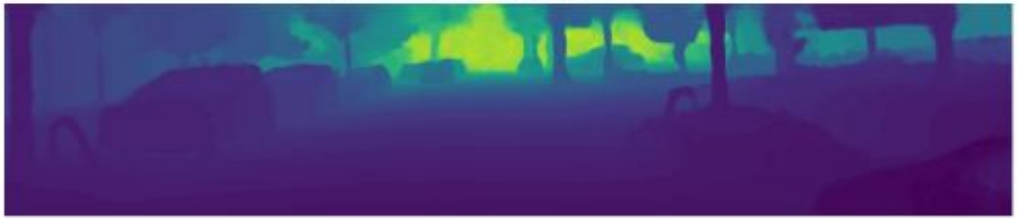
Yun Chen[1] Bin Yang[1,2] Ming Liang[1] Raquel Urtasun[1,2]
[1]Uber Advanced Technologies Group [2]University of Toronto
{yun.chen, byang10, ming.liang, urtasun}@uber.com

ICCV 2019

Monocular Camera	RGB	No depth information
Stereo Camera	Dense depth	Fail to sense depth Inaccurate depth
LiDaR	Accurate depth	Sparse depth



Dense Depth Output



Backgrounds

Continuous Convolution

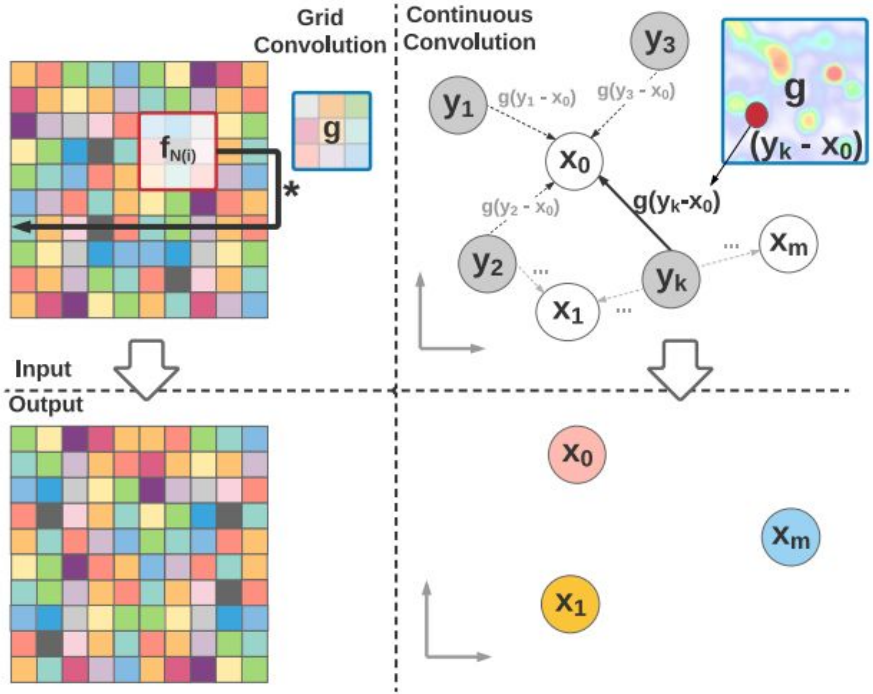


Figure 1: Unlike grid convolution, parametric continuous convolution uses kernel functions that are defined for arbitrary points in the continuous support domain. As a result, it is possible to output features at points not seen in the input.

Deep Parametric Continuous Convolutional Neural Networks

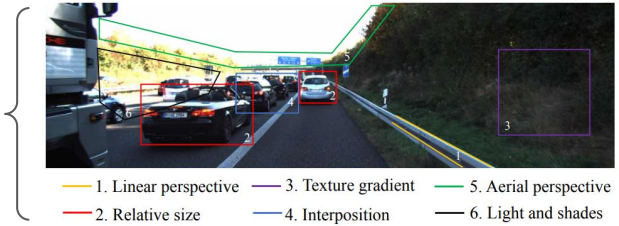
Shenlong Wang[1,3,*] Simon Suo[2,3,*] Wei-Chiu Ma[3] Andrei Pokrovsky[3] Raquel Urtasun[1,3]
[1]University of Toronto, [2]University of Waterloo, [3]Uber Advanced Technologies Group
{slwang, suo, weichiu, andrei, urtasun}@uber.com

$$h_{k,i} = \sum_{d}^{F} \sum_{j}^{N} g_{d,k}(\mathbf{y}_i - \mathbf{x}_j) f_{d,j}$$

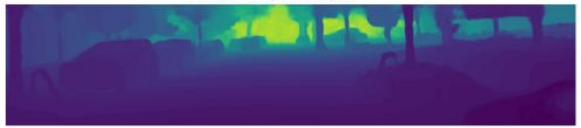
$$g(\mathbf{z}; \theta) = MLP(\mathbf{z}; \theta)$$

Depth estimation

RGB Image Input



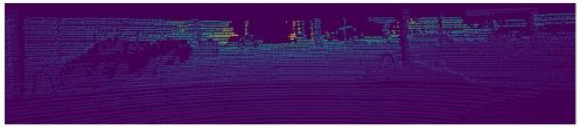
Dense Depth Output

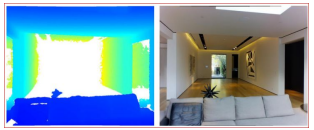


Depth completion

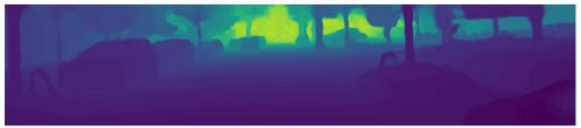
RGB Image Input

Sparse Depth Input



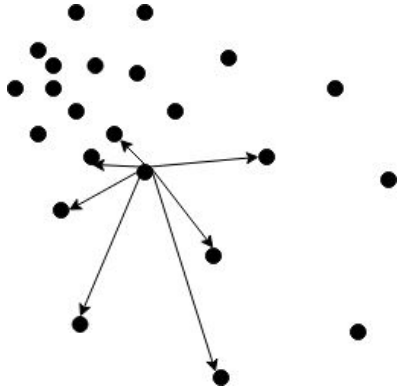


Dense Depth Output



Challenges on point clouds

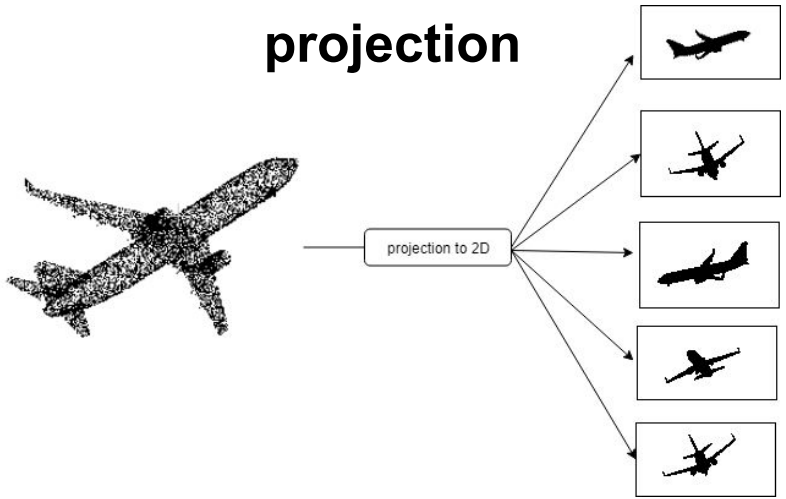
- Irregularity
- Unstructured
- Unorderdness



https://deepai.org/publication/review-deep-learning-on-3d-point-clouds

Projecting the 3D point cloud to 2D image space

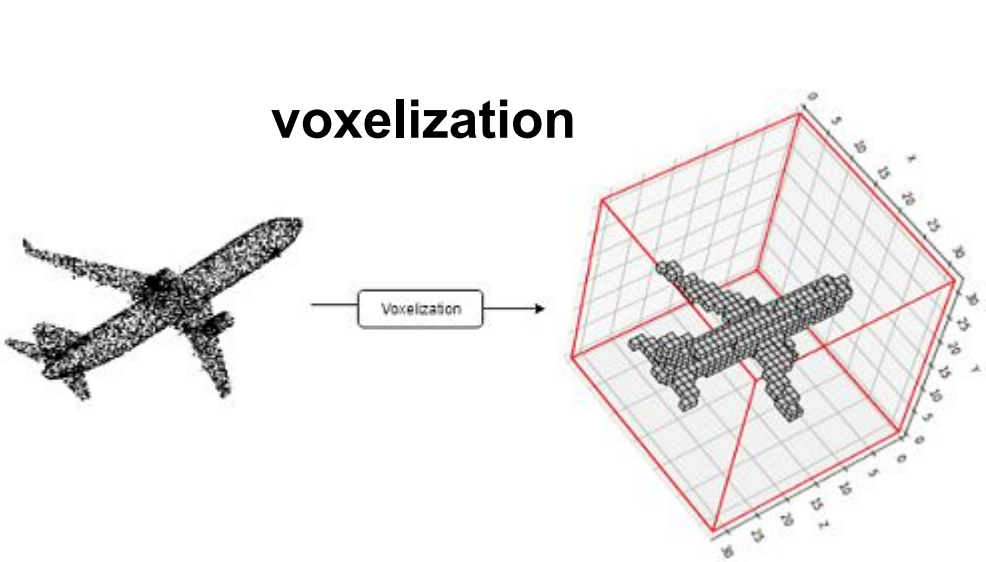
- Metric space is distorted
- Having difficulty capturing precise 3D geometric clues

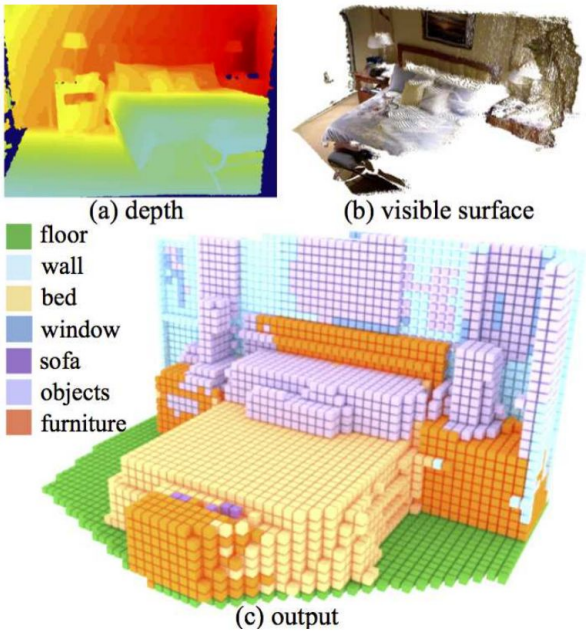


https://deepai.org/publication/review-deep-learning-on-3d-point-clouds

Producing a complete 3D voxel representation by applying 3D conv

- lack of exploitation of the dense image data, which can provide discriminative appearance clues

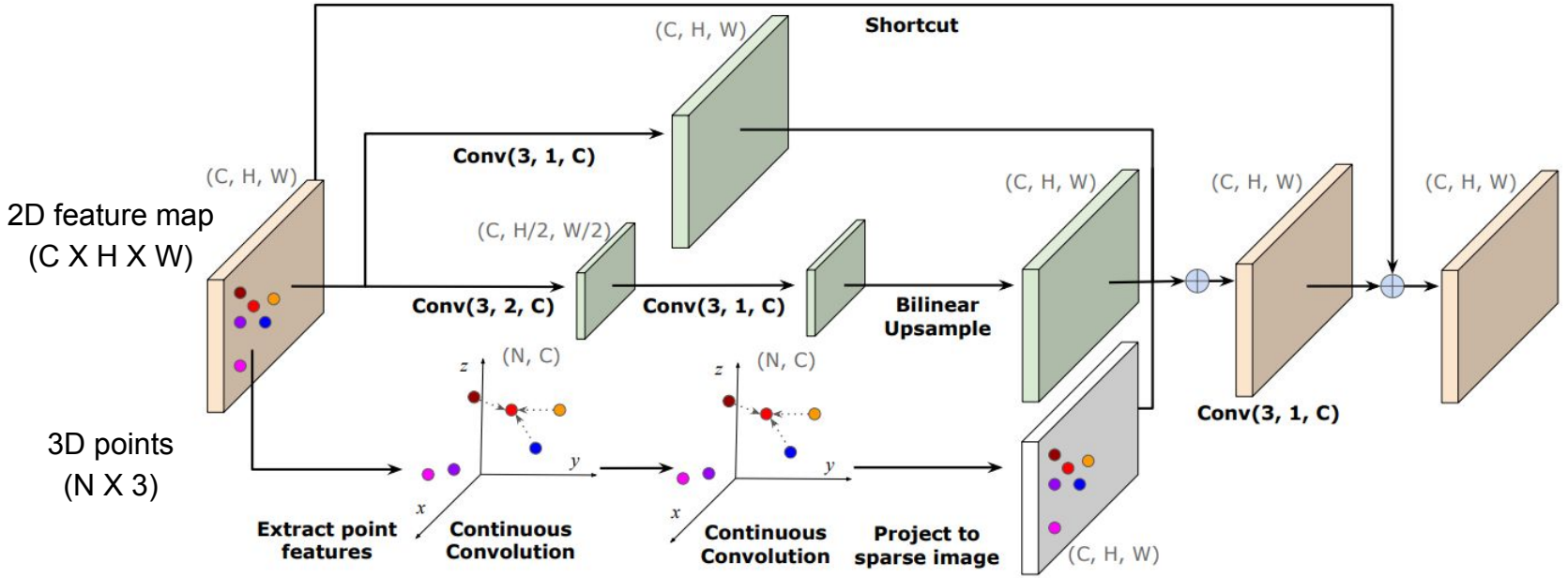




Semantic Scene Completion from a Single Depth Image (CVPR 2017)

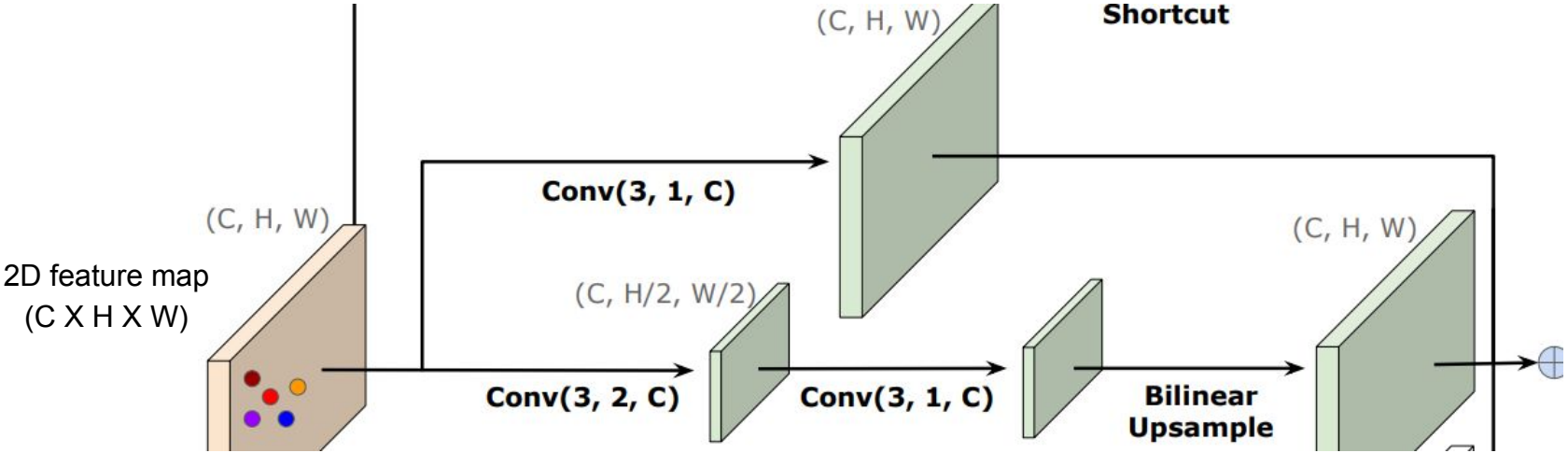
Taking advantage of both types of representations

- Extracting appearance features from dense pixels in 2D metric space
- Capturing geometric dependencies from sparse points in 3D metric space



Two-branch network structure to extract multi-scale features

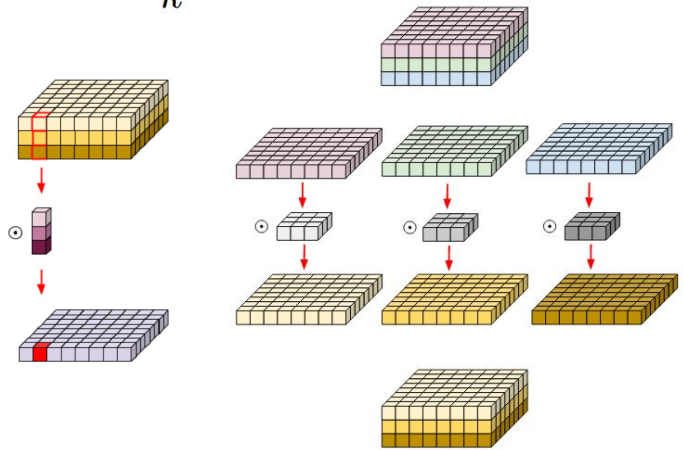
- Using 2D convolution to extract appearance features



Continuous conv directly on the 3D points to learn geometric features

- Output of each point is a weighted sum of transformed features of **neighbors**
  - non-grid, 3D points can be arbitrarily placed: K-nearest-neighbors
  - parametrizing the weighting function using a MLP

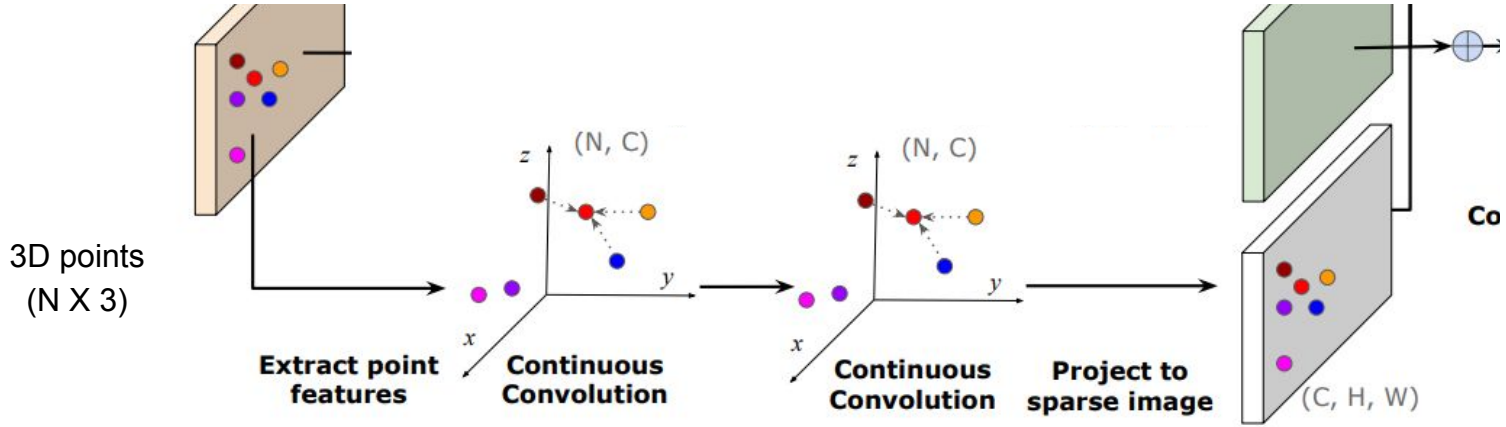
$$\mathbf{h}_i = W(\sum_k \mathrm{MLP}(\mathbf{x}_i - \mathbf{x}_k) \odot \mathbf{f}_k)$$



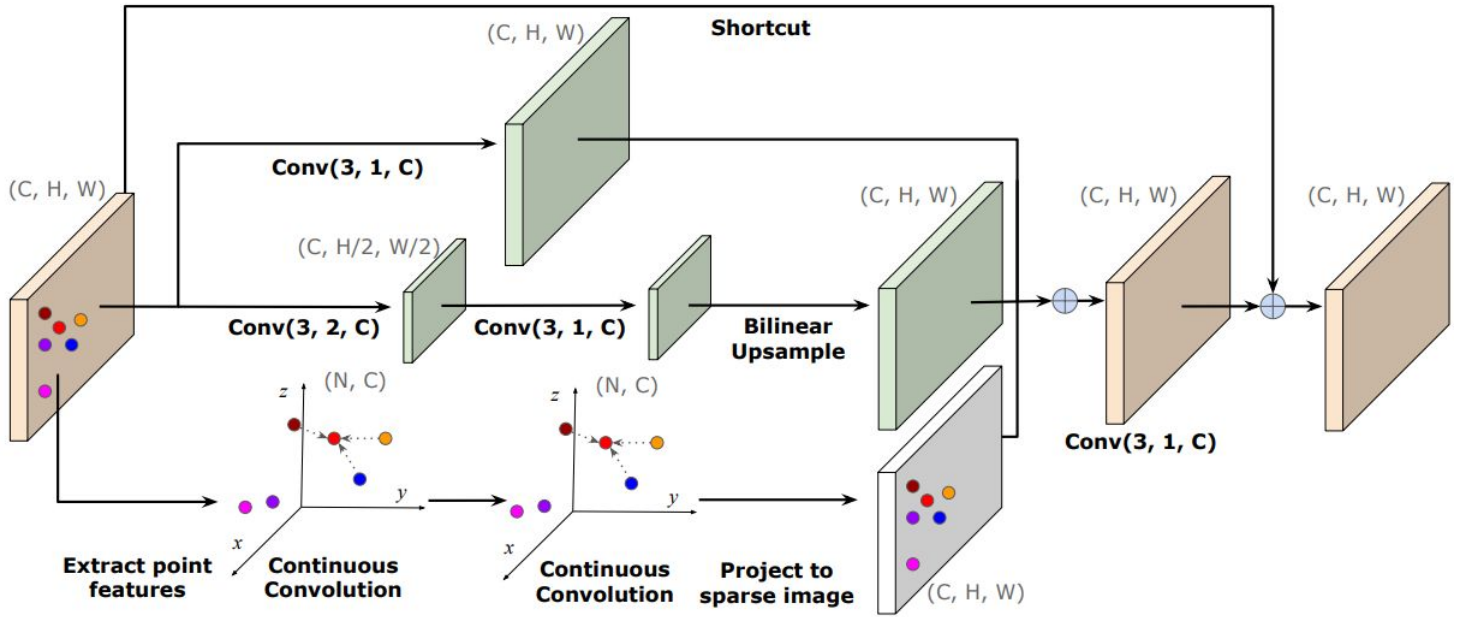
Continuous conv directly on the 3D points to learn geometric features

- output of each point is a weighted sum of transformed features of **neighbors**
  - non-grid, 3D points can be arbitrarily placed: K-nearest-neighbors
  - parametrizing the weighting function using a MLP

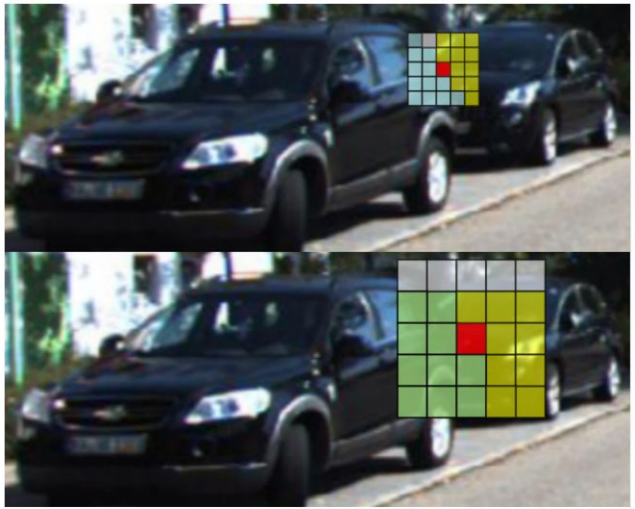
$$\mathbf{h}_i = W(\sum_k \text{MLP}(\mathbf{x}_i - \mathbf{x}_k) \odot \mathbf{f}_k)$$



Fusing the output feature maps of the 2D and 3D sub-networks



The fusing representation captures correlation in both spaces



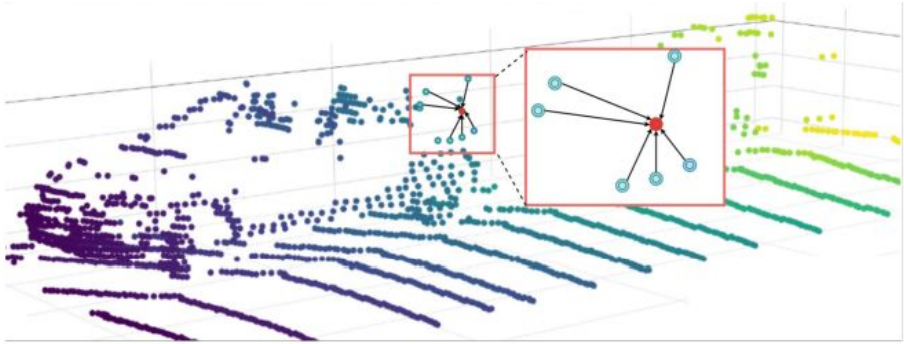


Figure 3. **Example receptive fields of conv(3, 1), conv(3, 2) and continuous convolution.** In 2D convolution, the neighbors are defined over image grids and are not necessarily close to each other in 3D space. The receptive field may cover both foreground and background objects. In the shown example convolution is performed at the red pixel. Green pixels are on the near car, and yellow pixels are on the distant car. In contrast, the neighbors in continuous convolution are based on the exact 3D geometric correlation.

Entire network architecture and learning

- Stacking 2D-3D fuse blocks into network
- Other approaches use multi-task objectives which leverage other tasks such as semantic segmentation to improve depth completion

$$\ell_1(d_i, l_i) = \begin{cases} 0.5(d_i - l_i)^2 & \text{if } |d_i - l_i| < 1 \\ |d_i - l_i| - 0.5 & \text{otherwise,} \end{cases}$$

$$\mathcal{L} = \ell_2 + \gamma \ell_1$$

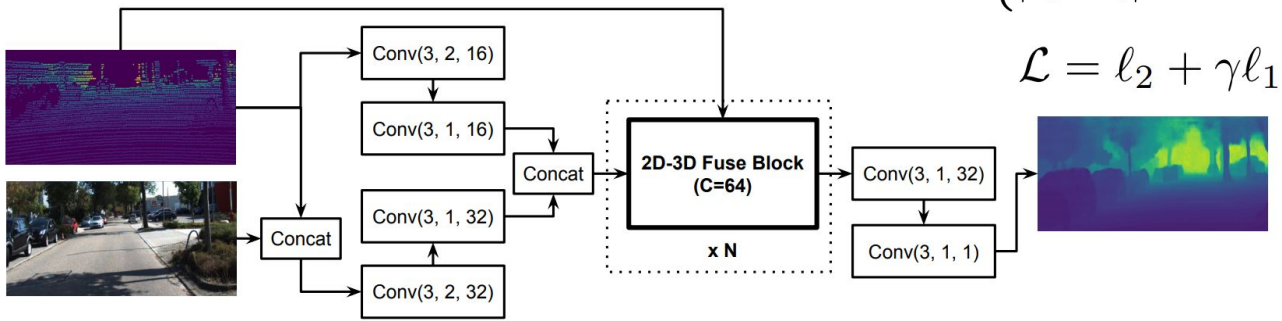


Figure 4. **Depth completion network based on 2D-3D fuse blocks.** The 2D-3D fused network takes image and sparse depth as input and predicts dense depth output. The main part of the network is the stacking of N 2D-3D fuse blocks. We also apply some convolution layers at the input and the output stage.

Entire network architecture and learning

- Dataset: KITTI depth completion
  - one sweep of LIDAR scan (sparse depth image), RGB image
  - ground truth: multiple sweeps of LIDAR scan (dense depth image)
- Evaluation Metrics: RMSE (Root Mean Square Error)

Method	**RMSE (mm)**	MAE (mm)	iRMSE (1/km)	iMAE (1/km)
SparseConvs [33]	1601.33	481.27	4.94	1.78
NN+CNN [33]	1419.75	416.14	3.25	1.29
MorphNet [4]	1045.45	310.49	3.84	1.57
CSPN [2]	1019.64	279.46	2.93	1.15
Spade-RGBsD [13]	917.64	234.81	2.17	0.95
NConv-CNN-L1 [7]	859.22	207.77	2.52	0.92
DDP† [39]	832.94	**203.96**	**2.10**	**0.85**
NConv-CNN-L2 [7]	829.98	233.26	2.60	1.03
Sparse2Dense [21]	814.73	249.95	2.80	1.21
DeepLiDAR† [26]	775.52	245.28	2.79	1.25
FusionNet† [34]	772.87	215.02	2.19	0.93
Our FuseNet	**752.88**	221.19	2.34	1.14

Table 1. Comparison with state-of-the-art methods on the test set of KITTI depth completion benchmark, ranked by RMSE. † indicates models trained with additional data and labels.

Receptive field

K nearest neighbors	3	6	9	12	15
RMSE	813	**810**	**810**	816	812

Table 2. Ablation study on number of nearest neighbors in the continuous convolution branch. Network config: $C = 32, N = 9$.

Architecture of the 2D-3D fuse block

stride_1 conv	stride_2 conv	cont. conv	RMSE (mm)
	✓	✓	840
✓		✓	826
✓	✓		817
✓	✓	✓	**803**

Table 3. Ablation study on the architecture of the 2D-3D fuse block. Network config: $C = 32, N = 12$.

Objective function

Loss	RMSE	MAE	iRMSE	iMAE
ℓ_2	790	232	2.51	1.16
smooth ℓ_1	839	**197**	**2.23**	**0.91**
ℓ_2, ℓ_2 + smooth ℓ_1	**785**	217	2.36	1.08

Table 4. Ablation study on objective function. Network config: $C = 64, N = 12$.

Network (width/depth) configuration: channel #, block

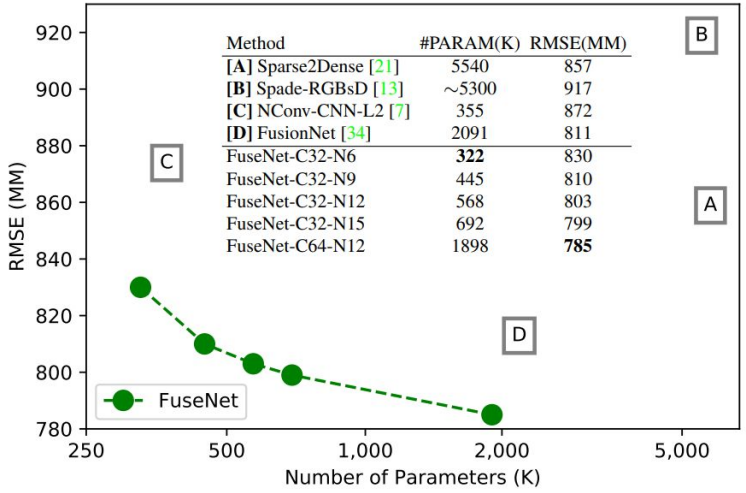
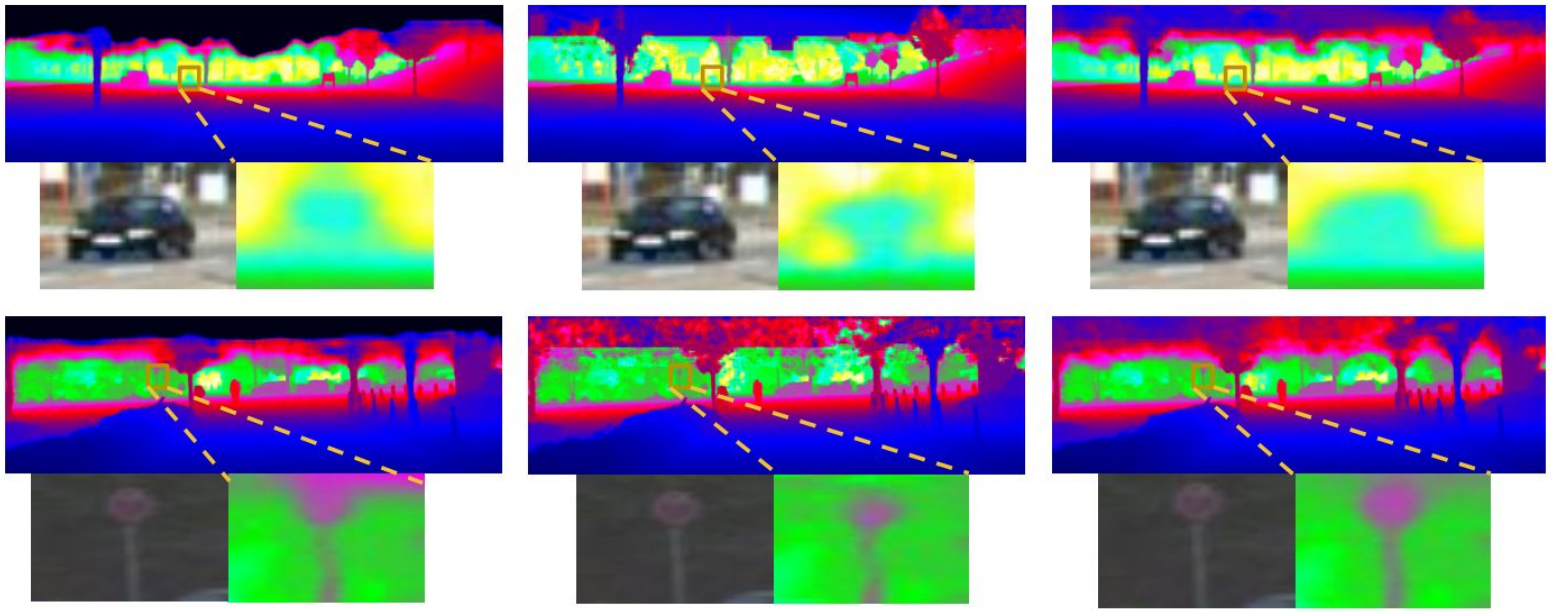


Figure 5. Trade-off between accuracy and model size by varying feature channel number C and block number N of the network.

Sharper boundaries of objects especially in the long range

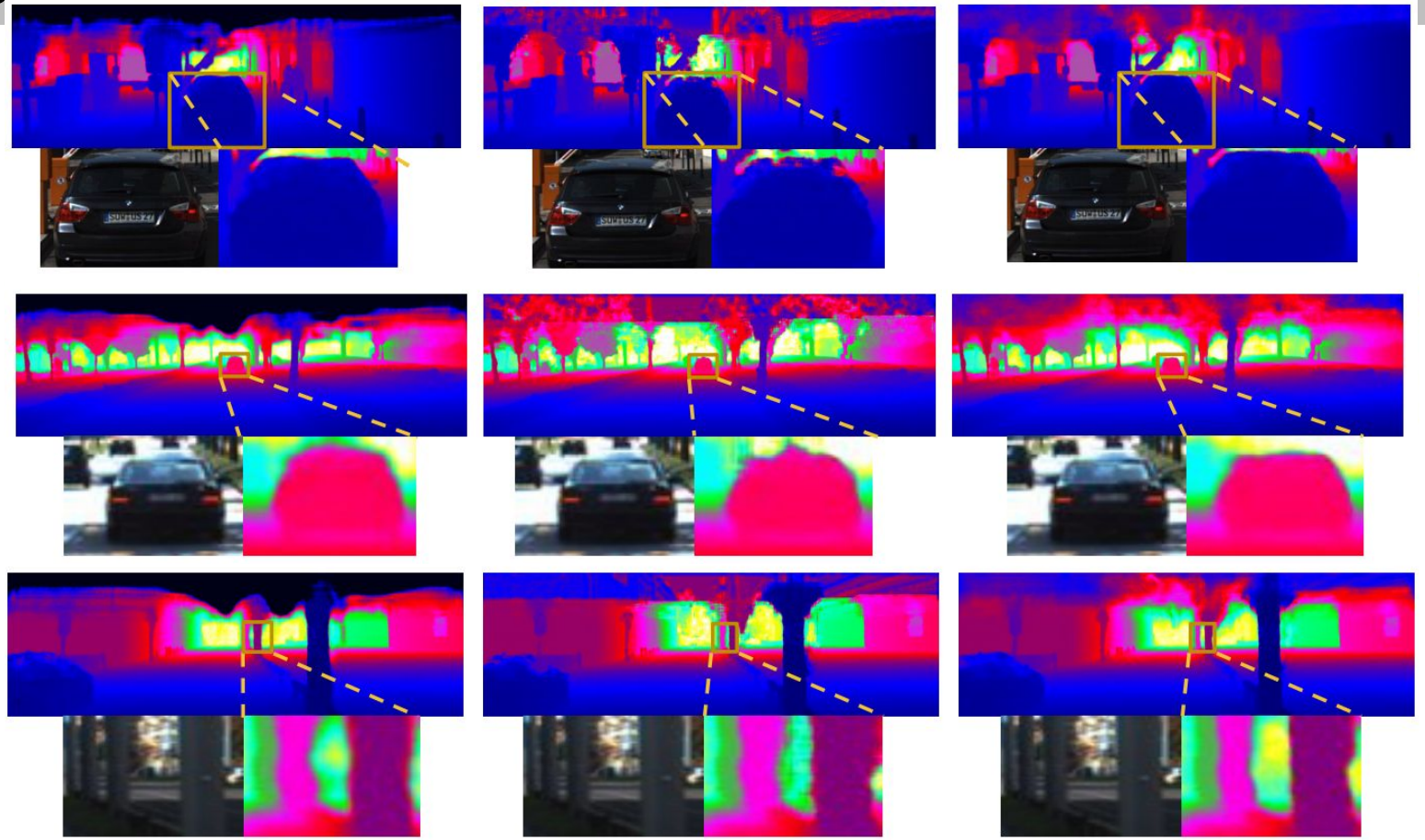
- Scale-invariant geometric feature



Sparse2Dense [21] NConv-CNN-L2 [7] Ours

Experiments

Qualitative Results



Sparse2Dense [21] | NConv-CNN-L2 [7] | Ours

Thank you!