Depth Completion Using Sparse LiDAR and RGB Inputs

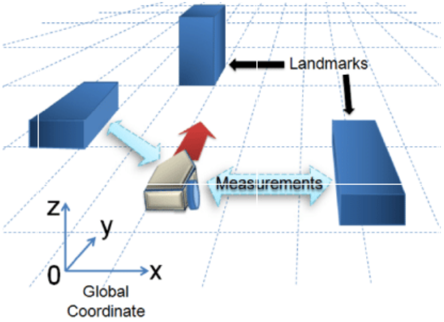
Alexander Fries

MSc Robotics, Cognition, Intelligence

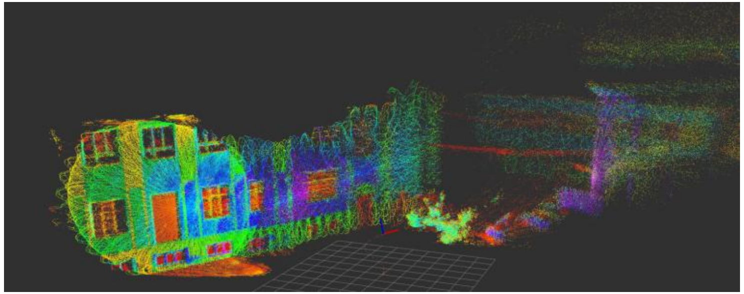
Munich, 02.12.2025

Introduction: What is Depth Completion (DC)?

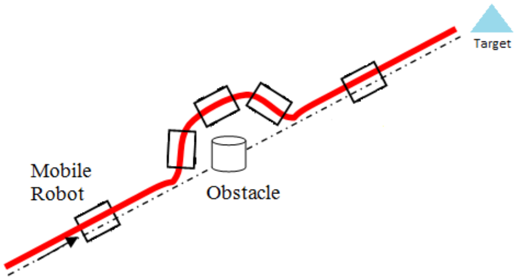
- **Definition:** Predicting a dense depth image from sparse and irregularly-spaced depth measurmements (e.g. LiDAR)
- **Why it matters:**



Localization



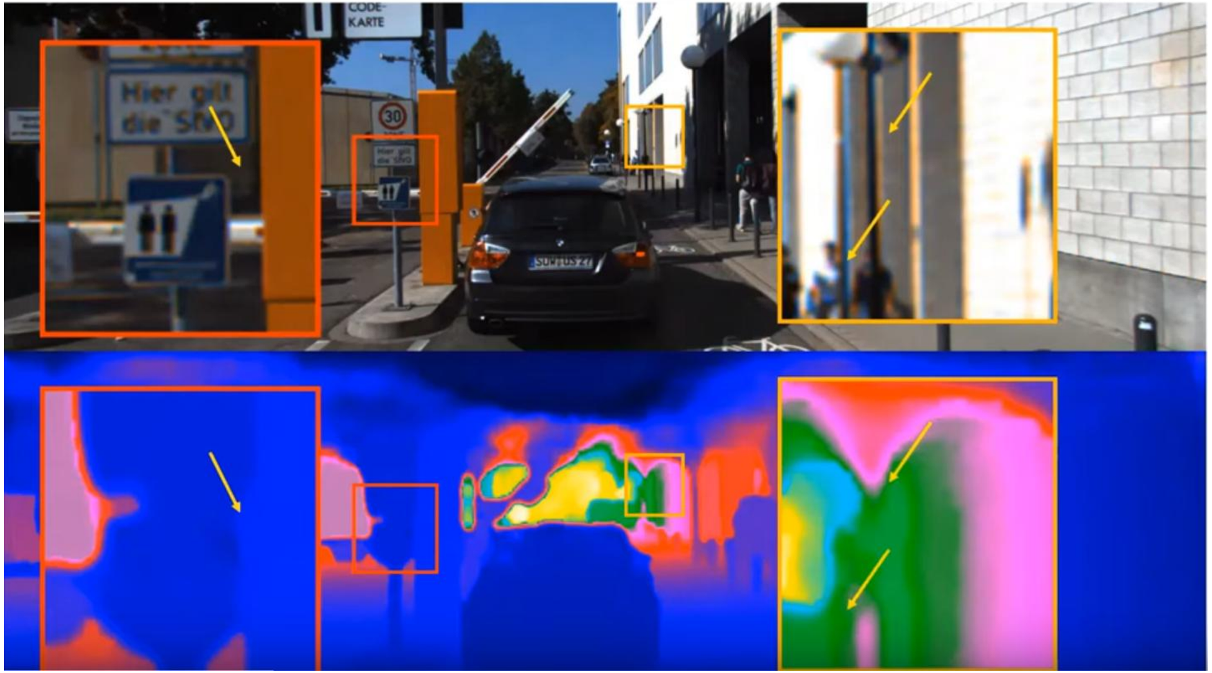
3D mapping



Obstacle avoidance

Introduction: Mixed-depth problem

Cheng *et al.*, "Depth estimation via affinity learned with convolutional spatial propagation network.", ECCV, 2018

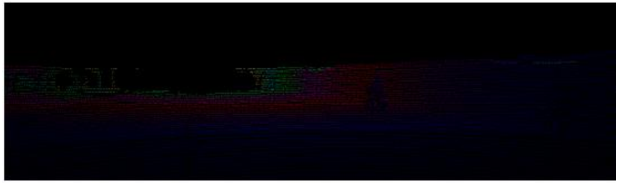


Introduction: Challenges in DC

LiDAR Data Limitatios (sparsity, spacing, costs)

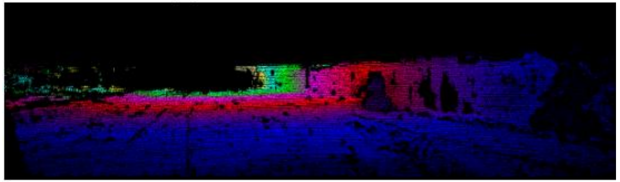
Handling multiple sensor modalities (LiDAR, RBG)

Ground Truth Availability



(a) raw LiDAR scans

(b) RGB



(c) semi-dense annotation

Outline

- Related Work
- Methods
- Experiments & Results
- Personal Comments
- Future Work
- Summary

1) *Self-Supervised Sparse-to-Dense Depth Completion* (Ma et al., 2018)
2) *Non-Local Spatial Propagation Network* (Park et al., 2020)

Related Work: An Overview

Traditional approaches (e.g. interpolation)

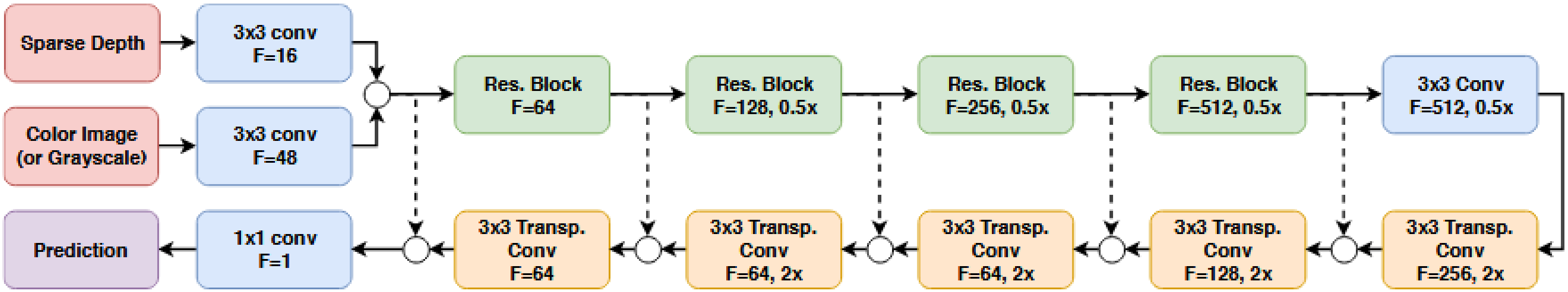
- **Focus:** Filling holes and removing noise in relatively dense depth maps
- **Drawbacks**:
  - Struggle with highly sparse data (e.g. LiDAR)
  - Fail to handle complex patterns near object boundaries

(Deep) Learning-based methods

- **Focus**: Leveraging RGB guidance and learned representations for sparse data
- Potential to address limitations of classical methods? **Let's find out!**

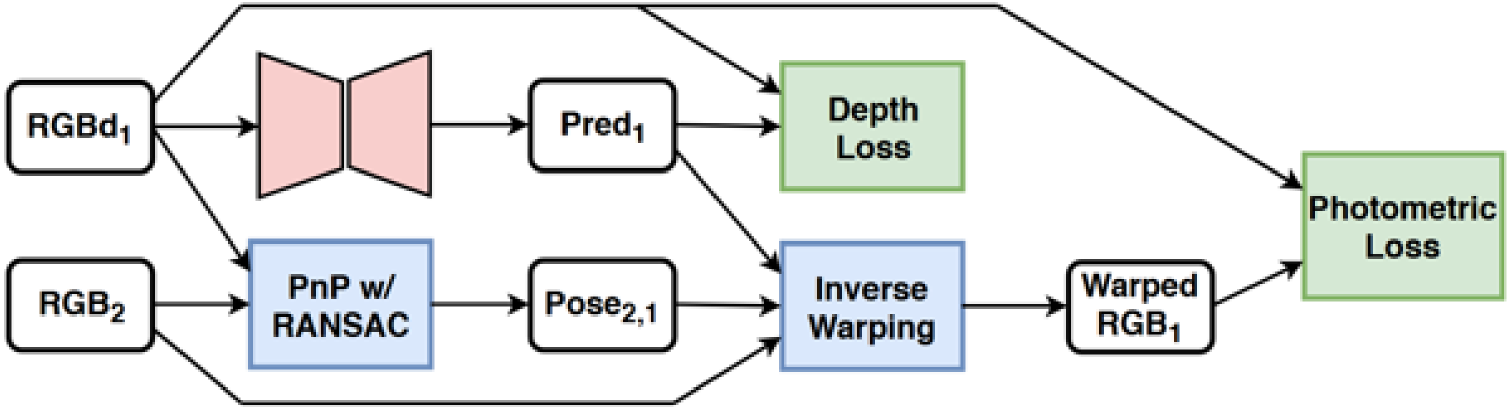
Method: Deep Regression Network for DC

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)



Method: Self-Supervised Training Framework

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)



Method: Losses

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)

$$\mathcal{L}_{\text{depth}}(\texttt{pred}, \texttt{d}) = \left\| \mathbb{1}_{\{\texttt{d}>0\}} \cdot (\texttt{pred} - \texttt{d}) \right\|_2^2$$

$$\mathcal{L}_{\text{photometric}}(\texttt{warped}_1, \text{RGB}_2) = \sum_{s \in S} \frac{1}{s} \left\| \mathbb{1}^{(s)}_{\{\texttt{d}==0\}} \cdot \left(\texttt{warped}_1^{(s)} - \text{RGB}_2^{(s)}\right) \right\|_1$$

$$\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{depth}}(\texttt{pred}_1, \texttt{d}_1) + \beta_1 \, \mathcal{L}_{\text{photometric}}(\texttt{warped}_1, \text{RGB}_1) + \beta_2 \, \left\| \nabla^2 \texttt{pred}_1 \right\|_1$$

Results: Data Set and Metrics

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)

Data set:

KITTI DC (for training and inference)

→Contains a semi-dense ground truth with ~30% annotated pixels

→No annotations in the top 1/3 of the images

KITTI Depth Completion Example

Error metrics:

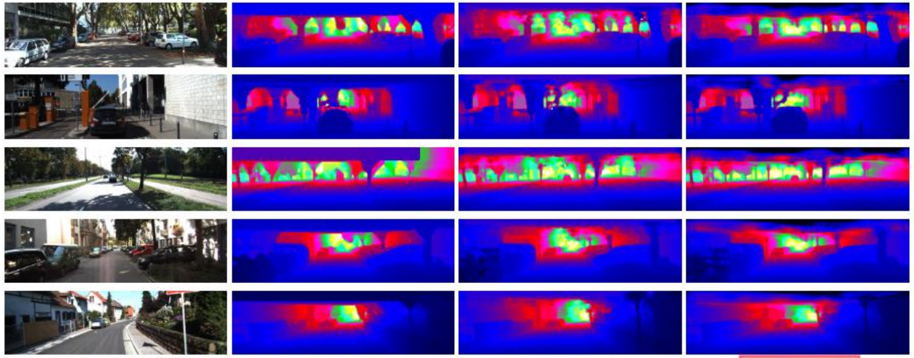
- RMSE (mm) : $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left| d_v^{gt} - d_v^{pred} \right|^2}$
- MAE (mm) : $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left| d_v^{gt} - d_v^{pred} \right|$
- iRMSE (1/km) : $\sqrt{\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left| 1/d_v^{gt} - 1/d_v^{pred} \right|^2}$
- iMAE (1/km) : $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left| 1/d_v^{gt} - 1/d_v^{pred} \right|$
- REL : $\frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}}\left| (d_v^{gt} - d_v^{pred})/d_v^{gt} \right|$
- δ_τ : Percentage of pixels satisfying $\max\left(\frac{d_v^{gt}}{d_v^{pred}}, \frac{d_v^{pred}}{d_v^{gt}}\right) < \tau$

Results: Comparison with SOTA Methods

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)

Table 1: Comparison against state-of-the-art algorithms on the test set.

Method	Input	rmse [mm]	mae [mm]	irmse [1/km]	imae [1/km]
NadarayaW [4]	d	1852.60	416.77	6.34	1.84
SparseConvs [4]	d	1601.33	481.27	4.94	1.78
ADNN [22]	d	1325.37	439.48	59.39	3.19
IP-Basic [20]	d	1288.46	302.60	3.78	**1.29**
NConv-CNN [21]	d	1268.22	360.28	4.67	1.52
NN+CNN2 [4]	d	1208.87	317.76	12.80	1.43
Ours-d	d	**954.36**	**288.64**	**3.21**	1.35
SGDU [18]	RGBd	2312.57	605.47	7.38	2.05
Ours-RGBd	RGBd	**814.73**	**249.95**	**2.80**	**1.21**



(a) RGB (b) NConv-CNN [21] (c) NN+CNN2 [4] (d) Ours-d

Results: Ablation Study (importance of net components)

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)

Table 2: Ablation study of the network architecture for depth input. Empty cells indicate the same value as the first row of each section. See Section 6.2 for detailed discussion.

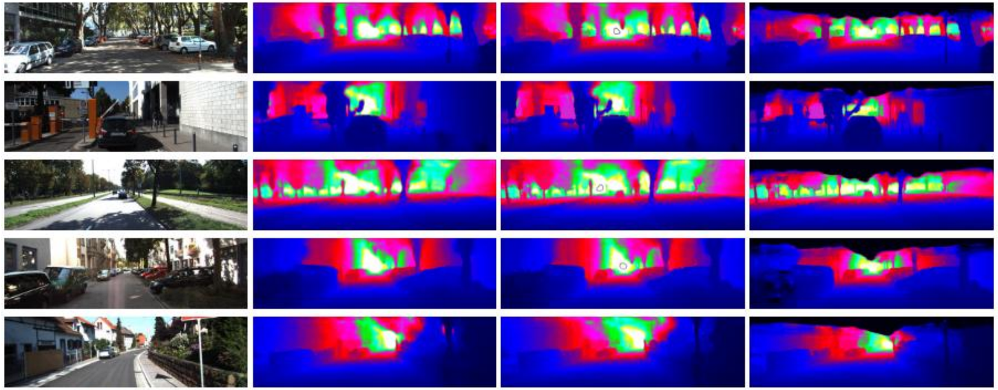
image	fusion split	loss	ResNet depth	with skip	reduced filters	pre-trained	№ pairs	down-sample	dropout & weight decay	rmse [mm]
None	-	L_2	34	Yes	2x ($F_1 = 32$)	No	5	No	No	**991.35**
		L_1								1170.58
			18							1003.78
				No						1060.64
					1x ($F_1 = 64$)					992.663
					1x ($F_1 = 64$)	Yes				1058.218
					4x ($F_1 = 16$)					1015.204
							4			996.024
							3			1005.935
								Yes		1045.062
									Yes	1002.431
Gray	16/48	L_2	34	Yes	1x ($F_1 = 64$)	No	5	No	Yes	**856.754**
RGB										859.528
	32/32									868.969
			18							875.477
				No						1070.789
	8/24				2x ($F_1 = 32$)					887.472
							4			857.154
							3			857.448
								Yes		859.528

Results: Evaluation of Self-Supervised Framework

1) Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera (Ma et al., 2018)

Table 3: Evaluation of the self-supervised framework on the validation set

Training Method	rmse [mm]	mae [mm]	irmse [1/km]	imae [1/km]
Photometric Loss Only	1901.16	658.13	5.85	2.62
Self-Supervised	1384.85	358.92	4.32	1.60
Supervised Learning	878.56	260.90	3.25	1.34



(a)RGB (b)Photometric Only (c)Self-supervised (d)Supervised

Pros:

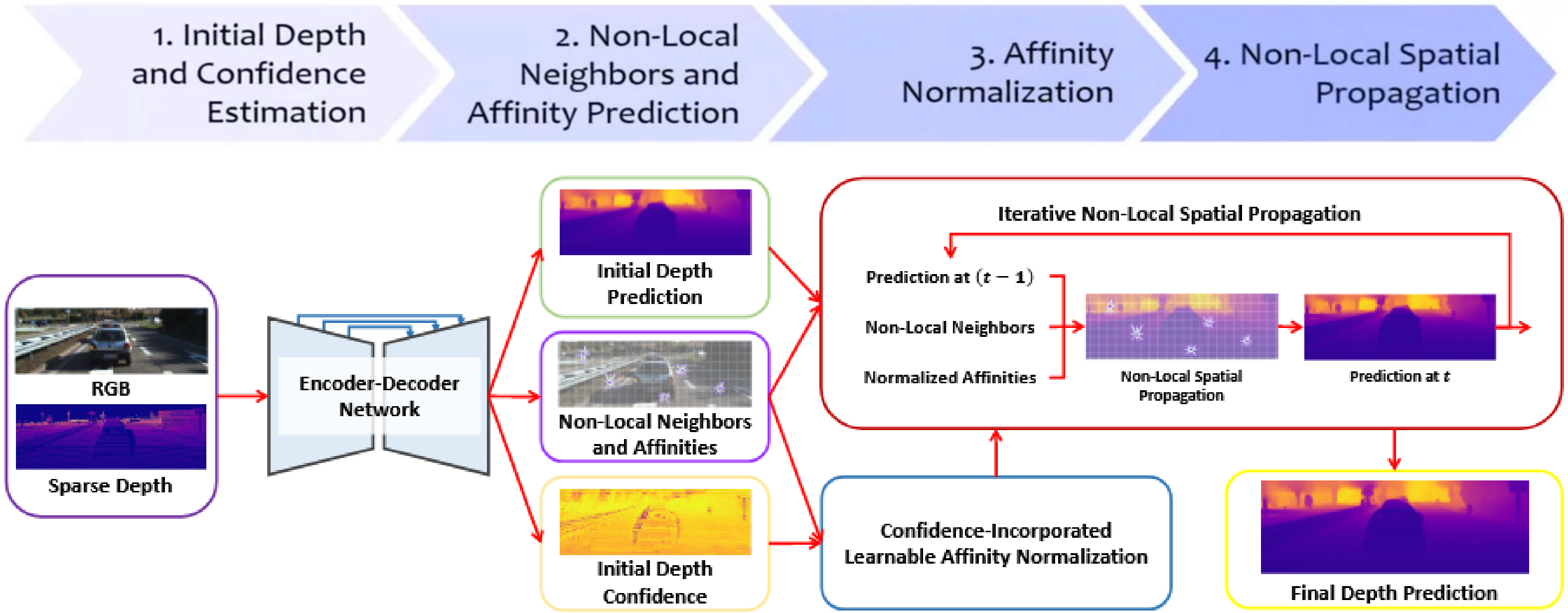
- Achieved SOTA resuts on error metrics
- NN architecture flexible

Cons:

- Does not consider dynamic objects (PnP algorithm may fail)
- Architecture optimized for 64-line LiDARs

Method: Algorithm and architecture

2) Non-Local Spatial Propagation Network (Park et al., 2020)



Method: Non-Local Spatial Propagation

2) Non-Local Spatial Propagation Network (Park et al., 2020)

Neighbors are predicted for each pixel (*i.e.*, Spatially varying)

$$\mathcal{N}^{\mathrm{NL}}_{m,n} = \{x_{m+p,n+q} \mid (p,q) \in f_{\phi}(\mathbf{I}, \mathbf{D}, m, n),\ p,q \in \mathbb{R}\}$$

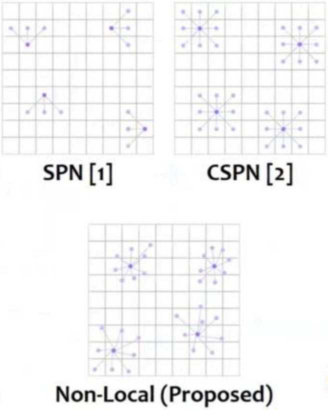
Learnable parameters

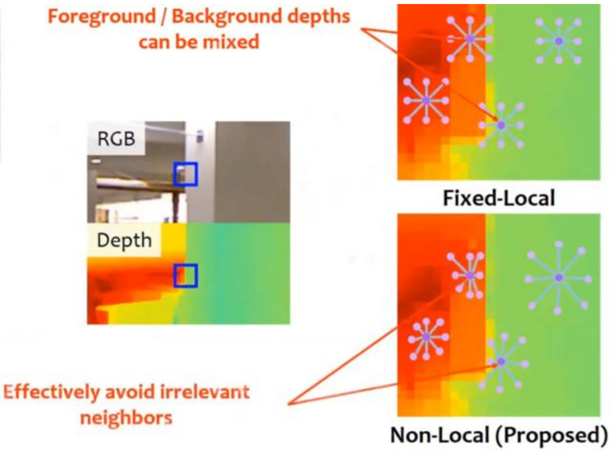
Offsets are real-valued (*i.e.*, sub-pixel)

For each pixel, its non-local neighbors are predicted.

$f_{\phi}(\cdot)$: Neighbor prediction function with learnable parameters ϕ

$\mathbf{I}, \mathbf{D}$: Input RGB and depth images





Method: Affinity Learning

2) Non-Local Spatial Propagation Network (Park et al., 2020)

$$x_{m,n}^{t} = \boxed{w_{m,n}^{c}} x_{m,n}^{t-1} + \sum_{(i,j)\in\mathcal{N}_{m,n}} \boxed{w_{m,n}^{i,j}} x_{i,j}^{t-1}$$

How much information should be propagated from neighbors?

$\mathbf{X} = (x_{m,n}) \in \mathbb{R}^{M\times N}$: A 2D map to be updated

$\mathcal{N}_{m,n}$: Neighbors of $x_{m,n}$

Method: Confidence-incorporated affinity normalization

2) Non-Local Spatial Propagation Network (Park et al., 2020)

- **Confidence scores** are predicted together with initial depth estimation.

- Affinity values are **weighted** by the corresponding **confidence score c_i**.

$$\tilde{w} = c_i \frac{\tanh(\hat{w}_i)}{\gamma}$$

Method: Loss function

2) Non-Local Spatial Propagation Network (Park et al., 2020)

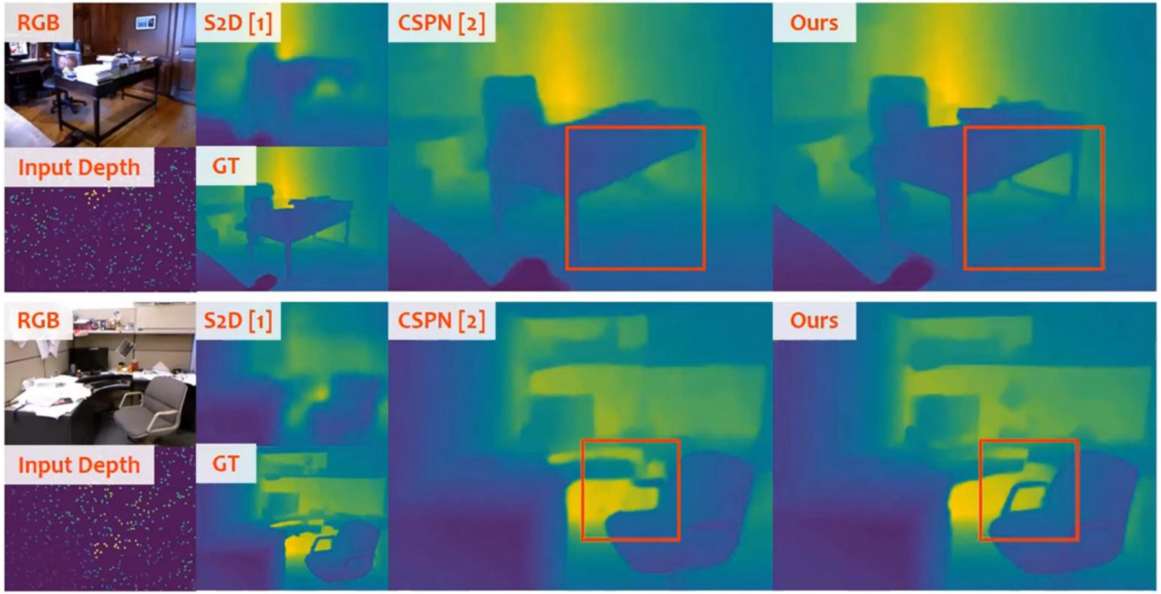
$$L_{recon}(\mathbf{D}^{gt}, \mathbf{D}^{pred}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} |d_v^{gt} - d_v^{pred}|^{\rho}$$

$\mathbf{D}^{gt}$: The ground truth depth
$\mathbf{D}^{pred}$: Prediction from the network
ρ : 1 for ℓ_1 and 2 for ℓ_2
$\mathcal{V}$: Valid pixels of $\mathbf{D}^{gt}$
$|\mathcal{V}|$: The number of valid pixels $\mathcal{V}$

- The entire net is trained end-to-end
- No direct supervision on non-local neighbors, affinities, and confidences

Results: NYU Depth V2

2) Non-Local Spatial Propagation Network (Park et al., 2020)

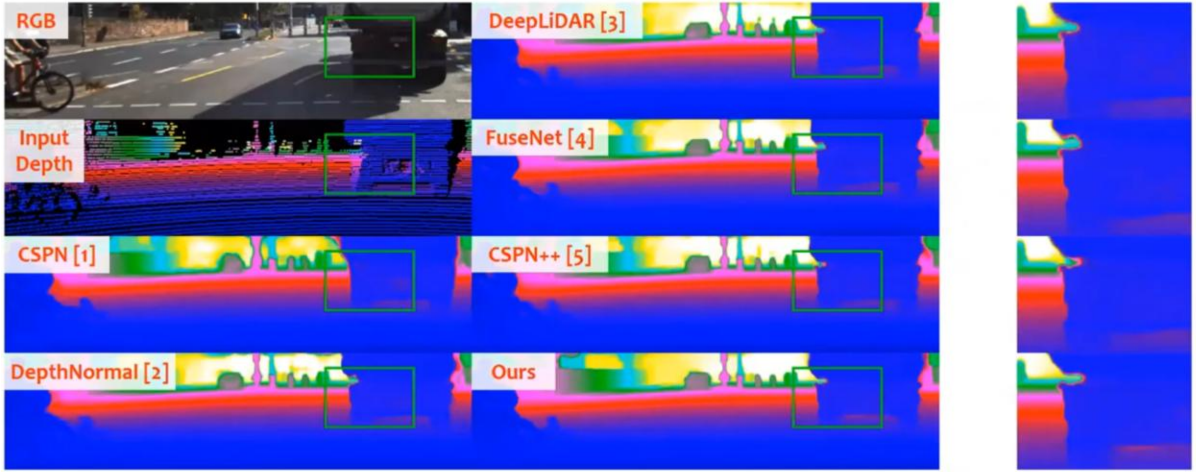


Method	RMSE (m)	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
S2D [21]	0.230	0.044	97.1	99.4	99.8
[21]+Bilateral [4]	0.479	0.084	92.4	97.6	98.9
[21]+SPN [19]	0.172	0.031	98.3	99.7	99.9
DepthCoeff [14]	0.118	0.013	99.4	**99.9**	-
CSPN [9]	0.117	0.016	99.2	**99.9**	**100.0**
CSPN++ [8]	0.116	-	-	-	-
DeepLiDAR [25]	0.115	0.022	99.3	**99.9**	**100.0**
DepthNormal [32]	0.112	0.018	99.5	**99.9**	**100.0**
Ours	**0.092**	**0.012**	**99.6**	**99.9**	**100.0**

Table 1. Quantitative evaluation on the NYUv2 [29] dataset. Results are borrowed from each paper. Note that S2D [21] uses 200 sampled depth points per image as the input, while the others use 500.

Results: KITTI Depth Completion

2) Non-Local Spatial Propagation Network (Park et al., 2020)



Method	RMSE (mm)	MAE	iRMSE	iMAE
CSPN [9]	1019.64	279.46	2.93	1.15
DDP [33]	832.94	203.96	2.10	0.85
NConv [12]	829.98	233.26	2.60	1.03
S2D [21]	814.73	249.95	2.80	1.21
DepthNormal [32]	777.05	235.17	2.42	1.13
DeepLiDAR [25]	758.38	226.50	2.56	1.15
FuseNet [7]	752.88	221.19	2.34	1.14
CSPN++ [8]	743.69	209.28	2.07	0.90
Ours	**741.68**	**199.59**	**1.99**	**0.84**

Table 2. Quantitative evaluation on the KITTI DC test dataset [30]. The results from other methods are obtained from the KITTI online evaluation site.

Results: Ablation Studies

2) Non-Local Spatial Propagation Network (Park et al., 2020)

Neighbors	Affinity	Normalization	Confidence	RMSE (mm)
Fixed-Local	Learned	Abs-Sum	-	908.4
Fixed-Local	Learned	Abs-Sum	Proposed	891.6
Fixed-Local	Learned	Tanh-γ-Abs-Sum*	-	896.4
Fixed-Local	Learned	Tanh-γ-Abs-Sum*	Proposed	890.4
Non-Local	Color	Tanh-γ-Abs-Sum*	Proposed	930.3
Non-Local	Learned	Abs-Sum	-	903.1
Non-Local	Learned	Abs-Sum	Proposed	889.5
Non-Local	Learned	Abs-Sum*	Proposed	886.0
Non-Local	Learned	Tanh-C	Proposed	886.4
Non-Local	Learned	Tanh-γ-Abs-Sum*	-	891.3
Non-Local	Learned	Tanh-γ-Abs-Sum*	Binary	892.9
Non-Local	Learned	Tanh-γ-Abs-Sum*	Weighted	884.8
Non-Local	Learned	Tanh-γ-Abs-Sum*	Proposed	884.1

Pros:

- Achieved SOTA resuts on error metrics (better than self-supervised regression approach)

Cons:

- Non-local spatial propagation comp. heavy
- Potentially slower inference times

Personal comments

Strengths

- SOTA benchmark performance on DC datasets
- → Improved Robustness and accuracy
- Reduced label dependency

Limitations

- Computational complexity (esp. non-local operations)
- Pose estimation dependency (e.g. using PnP with RANSAC, can be challenging in dynamic environments)
- Generalization challenges (can arise from e.g. texture-less or highly reflective surfaces, low-light scenarios, etc)

Future work

Data enhancement and availability

- Develop larger and more diverse datasets
- Improve scalable annotation methods
- Enhance data quality and standardization

Architectural and algorithmic improvements

- Integrate other NNs, e.g. transformers or GNNs
- Ability to deal with highly dynamic objects/envs

Multi-modal and temporal integration

- Incorporate radar, sonar, IMU data
- Utilize temporal data from successive frames

Summary

DC is critical for many applications that require dense and accurate depth maps

Learning-based methods effectively leverage RGB and sparse depth data to improve accuracy and depth density → but challenges remain in robustness and generalization

Future work: data quality, algorithmic improvements, and multi-modality

Q&A

Thanks for your attention!

Questions?

Further references:

1) Khan, M. A. U., Nazir, D., Pagani, A., Mokayed, H., Liwicki, M., Stricker, D., & Afzal, M. Z. (2022). A Comprehensive Survey of Depth Completion Approaches. *Sensors*, *22*(18), 6969. https://doi.org/10.3390/s22186969

2) Hu, J., Bao, C., Ozay, M., Fan, C., Gao, Q., Liu, H., & Lam, T. L. (2022). Deep Depth Completion from Extremely Sparse Data: A Survey. ArXiv. https://arxiv.org/abs/2205.05335

3) Xie, Z., Yu, X., Gao, X., Li, K., & Shen, S. (2024). Recent advances in conventional and deep learning-based depth completion: A survey. IEEE Transactions on Neural Networks and Learning Systems, 35(3), 3395–3415. https://doi.org/10.1109/TNNLS.2022.3201534