DVMN: Dense Validity Mask Network for Depth Completion

Laurenz Reichardt¹, Patrick Mangat¹ and Oliver Wasenmüller¹

Abstract—LiDAR depth maps provide environmental guidance in a variety of applications. However, such depth maps are typically sparse and insufficient for complex tasks such as autonomous navigation. State of the art methods use image guided neural networks for dense depth completion. We develop a guided convolutional neural network focusing on gathering dense and valid information from sparse depth maps. To this end, we introduce a novel layer with spatially variant and content-dependent dilation to include additional data from sparse input. Furthermore, we propose a sparsity invariant residual bottleneck block. We evaluate our Dense Validity Mask Network (DVMN) on the KITTI depth completion benchmark and achieve state of the art results. At the time of submission, our network is the leading method using sparsity invariant convolution.

I. INTRODUCTION

The need for sensor-driven environmental guidance is increasing. Autonomous cars, drones, or industrial automation solutions rely on a combination of different technologies to perceive their surroundings, make decisions, and overcome challenges. In the field of autonomous driving, the use of stereo cameras, radar sensors, and Light Detection And Ranging (LiDAR) sensors is common for environmental perception. Rotating LiDAR sensors create 3D depth maps, measuring distances by emitting laser pulses. The data density in the produced depth maps depends on the sensors amount of laser scan-lines. Typical LiDAR depth maps such as the one in Figure 1 have a high amount of unobserved space. Such sparse depth maps are insufficient for complex tasks such as autonomous navigation.

The completion of sparse depth maps is an ongoing field of research. Early approaches were based on handcrafted filters and algorithms manipulating the sparse data. Current state of the art solutions rely on neural networks to complete depth maps. Neural network depth completion is split into two areas: unguided completion using only depth input, and guided completion benefiting from additional camera input. Guided networks show the most promising results, as the dense image information provides valuable features which aid in the completion of depth maps. This guidance has also been used for the benefit of scene flow estimation [1].

We introduce a straightforward, yet effective Convolutional Neural Network (CNN) with a dual encoder-single decoder structure including skip connections, similar to U-Net [2]. Our contributions are the following:

1. A network architecture focused on gathering dense and valid data from sparse depth maps.
2. A novel layer to overcome sparse data, using spatially variant and content-dependent dilation to gather additional neighborhood information and to reduce the spatial degradation of filters.
3. Integrating sparsity invariant convolution into a bottleneck structure and taking into account the propagation of valid data.

II. RELATED WORK

A. State of the Art - Depth Completion

CNNs make up the majority of state of the art networks for depth completion, frequently utilizing encoder-decoder structures with skip connections [2]. Sparse input presents a challenge to the standard convolutional operation, leading to performance decreases of CNNs. As the filter moves along sparse input, its receptive field can have varying patterns and amounts of valid data, directly affecting the output value.

Uhrig et al. [3] introduce the KITTI depth completion benchmark and propose sparsity invariant convolution (SI-convolution). SI-convolution considers only observed input through normalization and weighing, using a binary validity mask. Similarly, binary masks have been used to filter invalid
values [4]. The MA-bottleneck block of Yan et al. [5] combines SI-convolution with a residual bottleneck block [6], as this block aids in gradient propagation, reduces parameters and saves computational costs. Furthermore they propose the MA-fusion module, effectively combining features at decoder skip connections while reintroducing binary validity information. Jaritz et al. [7] reason that deep CNNs can learn to overcome sparsity without masks and focus training with varying depth map densities.

Because binary validity masks saturate in deep networks, Eldesokey et al. [8], [9] use continuous confidence maps to propagate data reliability instead of validity throughout the entire network. Later networks apply confidence with surface normal prediction [10], with the combination of both guiding recurrent refinement [11]. These solutions train their normal prediction with synthetic data [10] or based on principal component analysis [11]. Others use confidence to fuse image and depth features, giving more weight to the modality with less uncertainty [12], [13], [14].

SPN [15] proposes learned affinity matrices from guidance input, followed by spatial propagation for the refinement of segmentation masks. Affinity based refinement also benefits completed depth maps [16], strengthening object alignment and reducing blurry results. Some methods use refinement conjointly with confidence maps [17], [18], [19]. However, spatial propagation only works within a fixed local neighborhood. Adding deformable convolution [17], learned non-local neighbor information [20], or trainable parameters into the recurrent refinement process [18] further improves accuracy. Schuster et al. [21] propose an image guided, sparsity-aware, convolutional module, with subsequent spatial propagation refinement. Sparsity-awareness is achieved through a binary validity mask.

Multi-scale information improves the capability of networks to overcome differently sized or deformed input. Various networks integrate Spatial Pyramid Pooling (SPP) [22] for depth completion [5], [16], [18], [23]. Atrous Spatial Pyramid Pooling (ASPP) [24] has been studied at the end of an encoder [16] or within residual blocks [13]. Li et al. [25] combine multiple networks, each using different resolutions of sparse input. Re-scaled input is also used for the fusion of 2D and 3D information [26].

B. State of the Art - Dilated Convolution

Dilated convolution [27] learns features at different scale, without changing the spatial size of feature maps, adding parameters, or computational burden. However, the filters effectiveness depends on the dilation rate. The "holes" in the dilated filter also produce a gridding effect, skipping over direct neighborhood information when calculating an output value [28]. The DeepLab series [29], [30], [24] and PSP [31] integrate dilated convolution to replace deeper pooling layers within a network, retaining spatial dimensions. DeepLab also introduced ASPP, using parallel dilated convolution to gather multi-scale information. The Stacked Dilated Convolution (SDC) layer uses different dilation rates in parallel to make up an entire network [32]. Qiao et al. [33] introduce the Switchable Atrous Convolution (SAC) layer, utilizing a single attention map to spatially combine the output from different dilation rates. Likewise, Li et al. [34] use attention within their layer to combine feature maps of different dilation rates in the channel dimension.

III. METHODS

The aforementioned depth completion methods broadly study depth refinement, predominantly using learned affinity maps and spatial propagation. Confidence propagation has also been applied extensively, in some networks combined with refinement. Current state of the art solutions reveal two research gaps. To the best of our knowledge, there has been no implementation of a SI-convolutional network specifically focusing on the amount of valid pixels gathered from sparse data. Moreover, while multi-scale information is common, there has been limited research using dilated convolution for the benefit of gathering additional data from depth maps.

Based on this review, we formulate two primary research goals: increasing valid information gathered from sparse data by focusing on dense validity masks, and further exploring dilated convolution in the context of depth completion.

A. Sparsity Invariant Convolution

Because of its fundamental relevance to our developments, it is necessary to review Sparsity Invariant Convolution [3], depicted in Figure 2 SI-Convolution aims to improve convolutional operation on sparse data. Depth maps are projected into 2.5D space and the location of observed pixels are encoded in a binary validity mask. This mask is used by SI-convolution to weight and normalize the elements within the filter. The output $f_{u,v}(x,o)$ of SI-convolution is given analogous to Uhrig et al. [3] by

$$f_{u,v}(x,o) = \frac{\sum_{i,j}^{k} o_{u+i,d,v+j,d} w_{i,j} + b}{\sum_{i,j}^{k} o_{u+i,d,v+j,d} + \epsilon}$$

(1)
B. Sparsity Invariant Switch Layer (SISL)

In 2.5D sparse depth maps, there are spatial areas where a convolution observes few or no valid pixels, degrading its effectiveness. In such instances, the output does not consider neighboring information or is invalid. This issue especially affects early convolution layers. SI-convolution focuses on the propagation of valid information, but not on increasing the amount of valid input. For this reason, learned feature maps continue to exhibit sparsity and scan-line patterns, until the validity mask saturates through propagation.

Thus, we propose the Sparsity Invariant Switch Layer (SISL, see Figure 3), enabling a spatially- and content-dependent increase of dilation rate, using a binary switch. This allows SISL to gather additional information and reduce the spatial degradation of filters. We specifically choose dilated convolution over a larger receptive field for its distinct benefits. Dilated convolution requires less parameters and by extent generalizes better, saves computational cost, and resamples its input without changing the output dimension. Moreover, dilated convolution enables weight sharing between parallel layers, which allows SISL to be integrated seamlessly into any existing SI-convolution.

The binary switch function considers the content in a convolution filter by observing the binary validity mask. We use a switch, since the validity mask encodes the precise location of information, compared to learning this information through attention. In the case of a 3×3 convolution, the switch increases the dilation rate only if all eight outside pixels are empty. We specifically choose eight empty pixels, so neighbor information is not skipped. This spatially variant dilation rate alleviates the gridding effect, because data is only resampled where no information in the receptive field is excluded. The content in the center pixel remains the same regardless of dilation and it is not considered by the switch.

The proposed binary switch function can be described by

\[
    f_{u,v}^S(o) = \min \left( \sum_{i,j=-k}^{k} o_{u+i,v+j} - o_{u,v}, 1 \right)
\]

with the undilated kernel size = (2k+1)², the validity mask \( o \), and the resulting switch-map \( f_{u,v}^S(o) \). The output of SISL is

\[
    \text{output} = f^S(o)f^{d=1}(x,o) + (1 - f^S(o)) f^{d=2}(x,o)
\]

with \( f^{d=1}(x,o) \) and \( f^{d=2}(x,o) \) as the output of SI-convolution (based on Equation 1) with dilation rates \( d \). The switch-map is also applied for validity mask propagation

\[
    \text{mask} = f^{o,d=1}(o) + (1 - f^{o,d=2}(o)) f^{o,d=2}(o)
\]

with \( f^{o,d=1}(o) \) and \( f^{o,d=2}(o) \) as the output validity mask of the convolutions, according to Equation 2. The effect of SISL on the validity mask is shown in Figure 4.

C. SI-Residual Bottleneck

Furthermore, we propose the SI-Residual Bottleneck, inspired by He et al. [6] and Yan et al. [5]. However, Yan et al. [5] limit mask propagation to down-sampling layers. Sequential use of the same validity mask zeroes valid output at the next layer. In contrast, our bottleneck considers mask
propagation, retaining valid information in the network and adding robustness to varying sparsity.

A bottleneck structure allows for deeper networks, reducing parameters and computational costs. Analogous to He et al. [6], our SI-Residual Bottleneck (Figure 5) uses three consecutive layers: a $1 \times 1$ convolution reduces the input channels (bottleneck width), followed by e.g. a $3 \times 3$ SI-convolution, before another $1 \times 1$ convolution resizes the features to the desired output channels. Afterwards, the residual is added. If used for expansion, the residual of the bottleneck is resized by another $1 \times 1$ convolution. For validity propagation, we use the convoluted binary mask which is utilized for weighing and normalization in the SI-operation, and limit it to a maximum value of one. The result is identical to Equation 2, however zero padding is not necessary. SI-convolution is not applied in the $1 \times 1$ convolutions, as these cannot observe neighboring information within their receptive field.

D. Dense Validity Mask Network

We name our final neural network Dense Validity Mask Network (DVMN), due to its focus on a dense validity mask. DVMN utilizes both SISL and the ”plain” SI-Residual Bottleneck, without weight sharing. The network is built as a dual encoder, single decoder structure, with skip connections. Its architecture is depicted in Figure 6.

1) Depth and Image Encoders: Both the depth and image encoders consist of four stages. Each stage expands the channels by $C$-stage using a SI-Residual Bottleneck, then down-samples using strided SI-convolution, followed by five subsequent SI-Residual Bottlenecks. This results in an encoder output of 128 feature maps at $\frac{1}{16}$ height and $\frac{1}{16}$ width. Due to the dense input, SI-convolution acts as a regular convolution in the image encoder. Both encoders are identical, but differ in one aspect: SISL is used in the first four bottlenecks of the first depth encoder stage.

The output of both encoders are fused with added multi-scale SPP context analogous to Yan et al. [5], reducing features at a ratio of 2:1.

2) Decoder: The decoder consists of four stages. The first three stages are identical, using transposed convolution to up-sample feature maps, and reducing the channel amount to $4C$ - $C$-stage. This is followed by skip connections using the MA-Fusion block [5] and a regular residual bottleneck. The decoder includes batch-normalization. The fourth encoder stage is a single transposed convolution, generating the completed depth map of original input dimensions.

IV. Evaluation

We perform the ablation and evaluation of our methods using the KITTI depth completion dataset [3]. The dataset contains sparse depth maps and aligned RGB-images, with splits of approximately 86K data-pairs for training, 1k for validation, and 1k for testing. Due to the LiDAR sensors vertical field of view, the top areas of the 2.5D depth maps do not contain measured nor ground-truth values. The sparse depth maps have approximately 5% observed data. For our ablation study we use the Root Mean Square Error (RMSE) and mean absolute error (MAE) error metrics. The benchmark uses the RMSE for ranking.

### Table I

| Version | Switch Design | Mask Prop. | Weight Sharing | d | Error Metric (mm) |
|---------|---------------|------------|---------------|---|------------------|
| v1      | Binary        | ✓          |               | 2 | 682.4            |
| v2      | Binary        | ✓          |               | 3 | 202.8            |
| v3      | Binary        | ✓          |               | 4 | 705.9            |
| v4      | Binary        | ✓          | ✓             | 2 | 696.4            |
| v5      | Binary        | ✓          | ✓             | 2 | 307.5            |
| v6      | None          | ✓          |               | 2 | 206.2            |
| -       | Binary E1 + Attention | ✓ | ✓ | 2 | 696.3 |
| -       | + Global Context |            | ✓ | 2 | 202.1 |
| -       | + Attention |            | ✓ | 2 | 682.4 |

### Table II

Ablation study comparing designs of our SI-Residual Bottleneck.
Revisiting [5] DVMN (ours)

Fig. 7. The output validity of the first seven layers of DVMN compared to Revisiting [5]. White pixels represent a valid output. The input layers (b) are downsized for this depiction, in the case of DVMN this is a SISL, followed by the down-sampling layers (c). DVMN uses three SISL and two SI-Residual Bottlenecks in (d), compared to five layers of Revisiting. The input image is depicted in (a). Since these validity masks are applied to learned feature maps in SI-convolution, our focus on a saturated validity mask reduces the scan-line pattern and sparsity in learned features.

For ablation we use the ADAM optimizer without weight decay, and the plateau learning rate schedule using a reduction factor of 0.5. The training duration is 50 epochs with an initial learning rate of 0.001 and a batch size of 4. For data augmentation we use horizontal and vertical axis flipping, random rotation, and added Gaussian noise. The loss function uses the mean squared error and adds smoothness loss scaled by the hyper-parameter \( \lambda = 0.1 \), to reduce gridded depth values [35]. We implement a dual encoder-single decoder structure with four stages and skip connections. Each encoder stage increases the channel dimension by \( C = 32 \). We use bottleneck widths of 0.5. The final network is described in III-D.

A. Ablation Studies

1) Sparsity Invariant Switch Layer: We remove individual components of SISL to verify their impact. We also study the layer with changed dilation rates and an attention switch. The results are shown in Table II.

In the first study we construct a depth decoder entirely from SISL (v1), excluding the down-sampling layers. The validity mask saturates with subsequent propagation, reducing the switch-activation. Through a mask analysis we determine that SISL impacts the first encoder stage. Beyond this initial stage, SISL continues to increases dilation in large areas of sparsity without a ground-truth, such as the top of depth maps or areas with measurement errors. For further ablation, we use SISL within the first depth encoder stage. Introducing weight sharing (v4) is detrimental to the network performance, indicating that SISL learns different weights for pixels further apart. SISL with a dilation rate of 2 shows the best performance. We believe this is because larger dilation rates skip over neighbor information. This skipped information becomes increasingly frequent as the validity mask saturates and consequently valid pixels are closer to a receptive field with dilation rate of two. We also remove the switch (v5), simply adding the results of the convolutions and output masks, with the results showing the benefit of content-depended dilation.

Removing the mask propagation (v5) and instead propagating validity exclusively at down-sampling layers, similar to Yan et al. [5], demonstrates the impact of mask propagation on performance. An example of the effect SISL has on the output validity mask is shown in Figure 4.

For deeper layers of SISL, when the validity mask is relatively saturated, we experiment with a spatial attention mechanism replacing the binary switch. This mechanism uses a \( 1 \times 1 \) convolution followed by the sigmoid activation function, to produce a spatial attention map. This SISL variant improves the networks accuracy, albeit at approximately 660,000 added parameters. Including global context inspired by Qiao et al. [33] worsens this result. This module summarizes information by channel-wise pooling and learns context through a \( 1 \times 1 \) convolution, adding the result to the original features. Most likely, a similar or better performance can be achieved by instead increasing parameters through network depth. SISL v1 will be chosen for our network.

2) SI-Residual Bottleneck: Inspired by further research from He et al. [36], we explore three bottleneck designs. Contrary to the original designs, the SI-convolution normalizes instead of batch normalization. The pre-activation design uses the activation functions before each convolutional layer. Pre-addition moves the last activation function before the added residual. For our ablation study we use the ReLU activation function. The results can be seen in Table I.

The developed "plain" SI-Residual Bottleneck (v1) significantly outperforms the MA-bottleneck design of Yan et al. [5], without increasing parameters or memory requirements. We attribute this performance increase to the efficient mask propagation. The "plain" bottleneck also outperforms its pre-activation and pre-addition variants.

B. Benchmark Evaluation

1) Quantitative Evaluation: DVMN was trained with the same regimen as the ablation studies, but using the AdamW optimizer with a weight decay of 0.01. Its performance was tested on the KITTI benchmark, achieving competitive results (refer to Table III). Our network ranks first among methods using SI-convolution, indicating that our focus on a
TABLE III
THE CURRENT KITTI BENCHMARK RANKINGS AT THE TIME OF WRITING.

| Method           | RGB-Input | Encoder-Decoder | Binary Mask | Confidence | Refinement | Surface Normals | Additional Dataset | Benchmark Results |
|------------------|-----------|-----------------|-------------|------------|------------|-----------------|-------------------|-------------------|
|                  |           |                 |             |            |            |                 |                   | iRMSE (1/km) | iMAE (1/km) | RMSE (mm) | MAE (mm) |
| PENet [19]       | ✓         | ✓               | ✓           | ✓          | ✓          |                 |                   | 2.17           | 0.94        | 730.08    | 210.55    |
| FCFR-Net [37]    | ✓         | ✓               | ✓           | ✓          |            |                 |                   | 2.20           | 0.98        | 735.81    | 217.15    |
| GuideNet [38]    | ✓         | ✓               | ✓           | ✓          |            |                 |                   | 2.25           | 0.99        | 736.24    | 218.83    |
| NLSNP [20]       | ✓         | ✓               | ✓           | ✓          |            |                 |                   | 1.99           | 0.84        | 741.68    | 199.59    |
| CSPN++ [18]      | ✓         | ✓               | ✓           | ✓          |            |                 |                   | 2.07           | 0.90        | 743.69    | 208.28    |
| ACMNet [14]      | ✓         | ✓               | ✓           | ✓          |            |                 |                   | 2.08           | 0.90        | 744.91    | 206.09    |
| DeepLidar [10]   | ✓         | ✓               | ✓           |            |            |                 |                   | 2.56           | 1.15        | 758.38    | 226.50    |
| MSG-CHN [25]     | ✓         | ✓               | ✓           |            |            |                 |                   | 2.30           | 0.98        | 762.19    | 220.41    |
| DSPN [17]        | ✓         | ✓               | ✓           |            |            |                 |                   | 2.47           | 1.03        | 766.74    | 220.36    |
| RGB&Uncertainty  | ✓         | ✓               | ✓           |            |            |                 |                   | 2.19           | 0.93        | 772.87    | 215.02    |
| DVMN (ours)      | ✓         | ✓               | ✓           |            | ✓          |                 |                   | 2.21           | 0.94        | 776.31    | 220.37    |
| PwP [11]         | ✓         | ✓               | ✓           |            |            |                 |                   | 2.42           | 1.13        | 777.05    | 235.17    |
| Revisiting [5]   | ✓         | ✓               | ✓           |            |            |                 |                   | 2.42           | 0.99        | 792.80    | 225.81    |
| CrossGuidance [13]| ✓         | ✓               | ✓           |            |            |                 |                   | 2.73           | 1.33        | 807.42    | 253.98    |
| NConv-CNN-L2 [9] | ✓         | ✓               | ✓           |            |            |                 |                   | 2.60           | 1.03        | 829.98    | 233.26    |
| SSGP [21]        | ✓         | ✓               | ✓           |            |            |                 |                   | 2.51           | 1.09        | 838.22    | 244.70    |
| NConv-CNN-L1 [9] | ✓         | ✓               | ✓           |            |            |                 |                   | 2.52           | 0.92        | 859.22    | 207.77    |
| Spade-RGBsD [7]  | ✓         | ✓               | ✓           |            |            |                 |                   | 2.17           | 0.95        | 917.64    | 234.81    |
| CSPN [16]        | ✓         | ✓               | ✓           |            |            |                 |                   | 2.93           | 1.15        | 1019.64   | 279.46    |
| Spade-sD [7]     | ✓         | ✓               | ✓           |            |            |                 |                   | 2.60           | 0.98        | 1035.29   | 248.32    |
| DFuseNet [23]    | ✓         | ✓               | ✓           |            |            |                 |                   | ✓              | 3.62        | 1.79      | 1206.66   | 429.93    |
| NConv [8]        | ✓         | ✓               | ✓           |            |            |                 |                   | ✓              | 4.67        | 1.52      | 1268.22   | 360.28    |
| ADNN [4]         | ✓         | ✓               | ✓           |            |            |                 |                   | ✓              | 59.39       | 3.19      | 1325.37   | 439.48    |
| SI-CNN [3]       | ✓         | ✓               | ✓           |            |            |                 |                   | ✓              | 4.94        | 1.84      | 1601.33   | 481.27    |

DVMN has a straightforward architecture, with a single depth map output. Other networks combine multiple depth map predictions into a final map, or refine their result in a second step. DVMN was trained without additional or synthetic data. Among the networks using confidence without refinement or additional data, only ACMNet surpasses DVMN, demonstrating that the performance of SI-convolutional networks can be comparable with confidence networks.

Our network with 2.16M trainable parameters is lightweight in relation to networks using ResNet-34 [6] or larger backbones [37], [18], [11], [16]. For reference, the original implementation of ResNet-34 has 21M parameters. Further improvements are conceivable by increasing the network depth or adding a refinement stage.

2) Qualitative Evaluation: To our knowledge, DVMN is the first network with focus on a dense validity mask. We compare the output validity of DVMN to Revisiting Sparsity Invariant Convolution [5], as a competitive network using SI-convolution, in Figure 7. Valid output is represented by a while pixel. The additional information gathered by our network is evident. We reach a comparable validity in three layers as Revisiting in its third encoder.

Furthermore, we compare our results to three other networks in Figure 8. In the first example our network (1d) achieves a good reconstruction of the barrier, but excels on the square sign before this barrier. Depth map (1c) produces a rounded sign. In the second example, DVMN has the best completion of the roof-line of the car (2d), with similar observations in (3d) and (4d).

V. CONCLUSION
In this paper we introduced the Dense Validity Mask Network, for image guided completion of sparse LiDAR depth maps. We focused on a dense validity mask, increasing the
amount of information available in our network. Specifically, we proposed a novel switch layer, using spatially variant and content-dependent dilatation to gather increased neighbor information and prevent filter degradation. We also integrated sparsity invariant convolution into a residual bottleneck structure, including validity propagation. Our network was evaluated on the KITTI depth completion benchmark and is currently the leading method using a SI-convolution.

VI. ACKNOWLEDGEMENTS

This work was funded by the Karl Völker Foundation in the project "KI-Fusion". We would also like to thank Dennis Teutscher for his support during the project.

REFERENCES

[1] R. Rishav, R. Battrawy, R. Schuster, O. Wasenmüller, and D. Stricker, “Deepslidarflow: A deep learning architecture for scene flow estimation using monocular camera and sparse lidar,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention (MICCAI), 2015.

[3] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, “Sparsity invariant cnns,” in International conference on 3D Vision (3DV), 2017.

[4] N. Chodosh, C. Wang, and S. Lucey, “Deep convolutional compressed sensing for lidar depth completion,” in Asian Conference on Computer Vision (ACCV), 2018.

[5] L. Yan, K. Liu, and E. Belyaev, “Revisiting sparsity invariant convolution: A network for image guided depth completion,” IEEE Access, 2020.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Computer Vision and Pattern Recognition (CVPR), 2016.

[7] M. Jaritz, R. De Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, “Sparse and dense data with cnns: Depth completion and semantic segmentation,” in International Conference on 3D Vision (3DV), 2018.

[8] A. Eldeeskey, M. Felsberg, and F. S. Khan, “Propagating confidences through cnns for sparse data regression,” arXiv preprint arXiv:1905.11993, 2018.

[9] ——, “Confidence propagation through cnns for guided sparse depth regression,” Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019.

[10] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, “Deepslidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[11] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, “Depth completion from sparse lidar data with depth-normal constraints,” in International Conference on Computer Vision (CVF), 2019.

[12] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, “Sparse and noisy lidar completion with rbg guidance and uncertainty,” in International Conference on Machine Vision Applications (MVA), 2019.

[13] S. Lee, J. Lee, D. Kim, and J. Kim, “Deep architecture with cross guidance between single image and sparse lidar data for depth completion,” IEEE Access, 2020.

[14] S. Zhao, M. Gong, H. Fu, and D. Tao, “Adaptive context-aware multi-modal network for depth completion,” arXiv preprint arXiv:2008.10833, 2020.

[15] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, “Learning affinity via spatial propagation networks,” in Advances in Neural Information Processing Systems (NeurIPS), 2017.

[16] X. Cheng, P. Wang, and R. Yang, “Depth estimation via affinity learned with convolutional spatial propagation network,” in European Conference on Computer Vision (ECCV), 2018.

[17] Z. Xu, H. Yin, and J. Yao, “Deformable spatial propagation networks for depth completion,” in 2020 IEEE International Conference on Image Processing (ICIP), 2020.

[18] X. Cheng, P. Wang, C. Guan, and R. Yang, “Cnsp++: Learning context and resource aware convolutional spatial propagation networks for depth completion,” in Association for the Advancement of Artificial Intelligence (AAAI), 2020.

[19] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, “Towards precise and efficient image guided depth completion,” arXiv preprint arXiv:2103.00783, 2021.

[20] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I.-S. Kweon, “Non-local spatial propagation network for depth completion,” in European Conference on Computer Vision (ECCV), 2020.

[21] R. Schuster, O. Wasenmüller, C. Unger, and D. Stricker, “SSgP: Sparse spatial guided propagation for robust and generic interpolation,” in Winter Conference on Applications of Computer Vision (WACV), 2021.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2015.

[23] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, “Disenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion,” in IEEE Intelligent Transportation Systems Conference (ITSC), 2019.

[24] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arXiv preprint arXiv:1706.05587, 2017.

[25] A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang, et al., “A multi-scale guided cascade hourglass network for depth completion,” in Winter Conference on Applications of Computer Vision (WACV), 2020.

[26] Y. Chen, B. Yang, M. Liang, and R. Urtasun, “Learning joint 2d-3d representations for depth completion,” in International Conference on Computer Vision (ICCV), 2019.

[27] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” arXiv preprint arXiv:1511.07122, 2015.

[28] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, “Understanding convolution for semantic segmentation,” in Winter Conference on Applications of Computer Vision (WACV), 2018.

[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs,” arXiv preprint arXiv:1412.7062, 2014.

[30] ——, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2017.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[32] R. Schuster, O. Wasenmüller, C. Unger, and D. Stricker, “Sdc-stacked dilated convolution: A unified descriptor network for dense matching tasks,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[33] S. Qiao, L.-C. Chen, and A. Yuille, “Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution,” arXiv preprint arXiv:2006.02334, 2020.

[34] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[35] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in Conference on computer vision and pattern recognition (CVPR), 2017.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in European Conference on Computer Vision (ECCV), 2016.

[37] L. Liu, X. Song, X. Lyu, J. Diao, M. Wang, Y. Liu, and L. Zhang, “Fcf-net: Feature fusion based coarse-to-fine residual learning for monocular depth completion,” arXiv preprint arXiv:2012.08270, 2020.

[38] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, “Learning guided convolutional network for depth completion,” IEEE Transactions on Image Processing, 2020.