The Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

Dynamic Spatial Propagation Network for Depth Completion

Yuankai Lin[1], **Tao Cheng**[2*], **Qi Zhong**[3], **Wending Zhou**[4] **and Hua Yang**[1]
[1]School of Mechanical Science and Enginering, Huazhong University of Science and Technology, China
[2]College of Urban Transportation and Logistics, Shenzhen Technology University, China
[3]Division of Logistics and Transportation, Tsinghua University, China
[4]School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), China
{linyuankai,huayang}@hust.edu.cn, chengtao@sztu.edu.cn, zhongq20@mails.tsinghua.edu.cn, 220019048@link.cuhk.edu.cn

Abstract

Image-guided depth completion aims to generate dense depth maps with sparse depth measurements and corresponding RGB images. Currently, spatial propagation networks (SPNs) are the most popular affinity-based methods in depth completion, but they still suffer from the representation limitation of the fixed affinity and the over smoothing during iterations. Our solution is to estimate independent affinity matrices in each SPN iteration, but it is over-parameterized and heavy calculation. This paper introduces an efficient model that learns the affinity among neighboring pixels with an attention-based, dynamic approach. Specifically, the Dynamic Spatial Propagation Network (DySPN) we proposed makes use of a non-linear propagation model (NLPM). It decouples the neighborhood into parts regarding to different distances and recursively generates independent attention maps to refine these parts into adaptive affinity matrices. Furthermore, we adopt a diffusion suppression (DS) operation so that the model converges at an early stage to prevent over-smoothing of dense depth. Finally, in order to decrease the computational cost required, we also introduce three variations that reduce the amount of neighbors and attentions needed while still retaining similar accuracy. In practice, our method requires less iteration to match the performance of other SPNs and yields better results overall. DySPN outperforms other state-of-the-art (SoTA) methods on KITTI Depth Completion (DC) evaluation by the time of submission and is able to yield SoTA performance in NYU Depth v2 dataset as well.

Introduction

Dense depth estimation is essential for 3D vision tasks, e.g., 3D object detection (Ma et al. 2020b) and tracking (Hu et al. 2019), 6D object pose estimation (Wang et al. 2019), simultaneously localization and mapping (SLAM) (Zhang and Singh 2015), and structure-from-motion (SfM) (Zhu et al. 2018), which are widely used in autonomous driving and robot perception. However, the output of depth sensors such as LiDAR and ToF cameras is sparse, resulting in a large number of empty data regions, which need further processing to fulfill or simply ignore these missing regions.

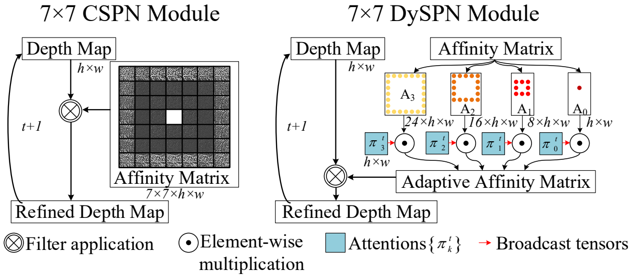

Figure 1: Comparison between 7×7 CSPN and 7×7 DySPN. Left: A fixed affinity matrix is assigned to CSPN, with its weights distributed in the outermost part of the kernel. Right: An adaptive affinity matrix adjusted by the attention mechanism is provided to our DySPN.

Since RGB images can reflect subtle changes of color and texture, many existing works (Ma and Karaman 2018) (Van Gansbeke et al. 2019) (Eldesokey, Felsberg, and Khan 2020) utilize synchronous RGB images as guidance to complete sparse depth values, which are known as image-guided depth completion. These methods directly estimate dense depths by RGB images and sparse depths and use encoder-decoder architecture (Badrinarayanan, Kendall, and Cipolla 2017) to generate more details of depth maps. Therefore, compared with the early methods (Hawe, Kleinsteuber, and Diepold 2011) (Uhrig et al. 2017) (Ku, Harakeh, and Waslander 2018) that only rely on sparse measurement, the results of depth completion are significantly improved. In order to generate better depth image boundaries, many learnable edge-preserving strategies (Cheng, Wang, and Yang 2020) (Tang et al. 2021) are adopted, which can be easily embedded into end-to-end deep convolutional neural networks. SPNs are based on the strategy of Anisotropic Diffusion (Weickert 1998), which are popular refinement techniques in depth estimation and depth completion. This strategy obtains the final dense depth maps by affinities of local (Cheng et al. 2020a) or non-local (Park et al. 2020) neighbors and iteratively refines the depth prediction. However, most of the current SPNs are based on linear propagation models. The values of the affinity matrix would not change during the propagation process, which greatly limits

---

[*] Corresponding author

the representation capability of SPNs. Taking CSPN (Cheng, Wang, and Yang 2018) as an example, as illustrated in Fig. 1, when the kernel size is 7, most of the affinity weights are distributed in the distance, which means that the affinity matrix cannot clearly represent the relationship between pixels and neighbors.

To solve the problems mentioned above, inspired by Dynamic filters (Jia et al. 2016), we proposed a new non-linear propagation design, named Dynamic Spatial Propagation Network (DySPN), which can learn an adaptive affinity matrix. To implement this design, different attention levels are assigned to neighbors of different distances when refining each pixel in the depth map recursively. More specifically, at the beginning of the propagation, far neighbors provide long-range information to fill the hole of the initial depth map and smooth it quickly. As the depth map becomes denser, near neighbors are paid more attention to edge-preserving. Our attention mechanism adjusts the affinity matrix according to each propagation stage, making the process more accurate and efficient. Besides, we adopt a diffusion suppression operation, which estimates the affinities of refining depths in every step to preserve good parts of the depths and avoid over-smoothing. The affinity of refining depth is easy to participate in the propagation process with nearly no additional computational cost. Although the computational complexity of our DySPN is similar to CSPN, we try to use a larger receptive field and fewer neighbors to achieve a better result. Thus, we introduce three variations that reduce the amount of distant pixels needed dramatically while still retaining similar accuracy. Compared with other SPNs, the iteration steps of our proposed method is only half, which is 6 in this paper.

In summary, our contribution lies in three aspects:

- We propose a non-linear propagation model (NLPM) named DySPN. This model adjusts affinity weights dynamically which is a new prospect to improve SPN. We further achieve it by neighborhood decoupling and spatial-sequential attention.
- A diffusion suppression (DS) operation is proposed, which can adaptive terminate the propagation and reduce over-smoothing.
- Three implementations of the DySPN model are proposed, which can effectively reduce computational complexities and improve performance.

Related Work

Image-guided depth completion Early depth completion methods (Hawe, Kleinsteuber, and Diepold 2011) (Yang, Kim, and Park 2012) (Chiu, Leou, and Hsiao 2014) only rely on sparse measurements to reconstruct or fill holes. image-guided depth Completion methods (Yang et al. 2007) (Park et al. 2011) (Ma and Karaman 2018) usually achieve better results because the additional RGB images contain a large amount of surface, edge, or semantic information.

Later works conduct the guidance or fusion strategies of these RGB image information. For example, DeepLiDAR (Qiu et al. 2019) is a network composed of a color pathway and a surface normal pathway. ACMNet (Zhao et al. 2021) adopts the graph propagation to capture the observed contextual information in the encoder stage. Inspired by the guided image filtering, GuideNet (Tang et al. 2021) proposes an image guidance convolution module for multi-modal feature fusion. FCFR-NET (Liu et al. 2021) designs an energy-based fusion operation to fuse the features of color and depth information. PENet (Hu et al. 2021) uses addition operation to merge color and depth features of different modalities at multiple stages. Besides, KBNet (Wong and Soatto 2021) transfers spatial positional encoding across different branches of the encoder by a calibrated backprojection module.

These methods usually use two independent branches to extract the features of RGB images and sparse depth maps and fuse them at different scales.

Spatial propagation networks SPNs are depth refinement technologies that differ from the aforementioned two-branch architectures. They are usually connected by a simple Unet (Ronneberger, P.Fischer, and Brox 2015) network. Among them, SPN (Liu et al. 2017) is the first proposed method to use a deep CNN to learn an affinity matrix that effectively models any pairwise similarity matrix. SPN builds a row/column linear propagation model whose spatially varying transformation matrix constitutes the affinity matrix. It can be easily applied to many high-level vision tasks, including depth completion and semantic segmentation. Later, CSPN (Cheng, Wang, and Yang 2018) further improves the linear propagation model and adopts a recursive convolution operation to be more efficient. CSPN++ (Cheng et al. 2020a) merges the outputs of three independent CSPN modules so that its propagation learns adaptive convolutional kernel sizes and the number of iterations. NLSPN (Park et al. 2020) predicts the relevant non-local neighbors to avoid the mixed-depth problem caused by irrelevant local neighbors. DSPN (Xu, Yin, and Yao 2020) utilizes their deformable SPN that adaptively adjusts receptive fields at each pixel. ODE-CNN(Cheng et al. 2020b) adopts an inverse gnomonic projection CSPN and a deformable CSPN, which better recovers the structural details of Omnidirectional depth maps. PENet (Hu et al. 2021) implement a dilated CSPN++ to enlarge the propagation neighborhoods.

However, these methods are constrained by the bottleneck of the linear propagation model, as the affinity matrix is fixed during propagation. Fixed affinity weights make it difficult for propagation to take into account both long-range dependencies and local information, resulting in more SPN iterations.

Dynamic filters Dynamic filters (Jia et al. 2016) (Su et al. 2019) could dynamically modify or predict filter weights based on the input features, and has shown a strong connection to the attention mechanism (Xu and Saenko 2016). CondConv (Yang et al. 2019) parameterizes several convolutional kernels as a linear combination to mix the experts voted. WeightNet (Ma et al. 2020a) directly generates the weights by a grouped fully-connected layer which is different from common practice that applies operations on vectors in feature space. DyNet (Zhang et al. 2020) presents a dynamic convolution kernel generation method whose coef-

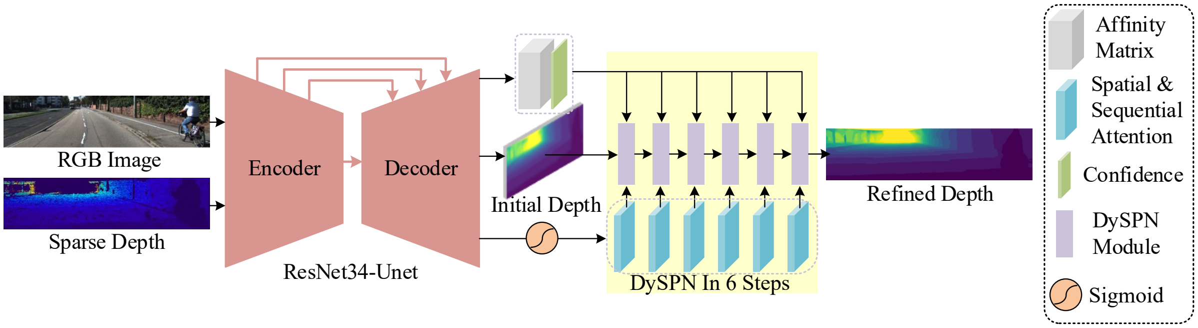

Figure 2: The overall framework of our networks with DySPN. We use a simple ResNet34-Unet to generate an initial depth map, an affinity matrix, a serial of spatial and sequential attention maps. The attention maps are split into six parts and applied to DySPN module in a 6-steps propagation process.

ficient prediction module is trainable and designed to predict the coefficients of fixed convolution kernels. DynamicConv (Chen et al. 2020) utilizes an Attention mechanism to aggregate multiple convolution kernels to increase the model capability. Furthermore, DDF (Zhou et al. 2021) simultaneously applies spatial and channel-wise attention to the dynamic filter of each pixel.

Inspired by the dynamic filters described above, we propose a DySPN with a non-linear propagation model, which applies spatial and sequential attention to generate a series of adaptive affinity matrices.

Our Approach

In this paper, we propose DySPN for image-guided depth completion, including a naive non-linear propagation model, a diffusion suppression operation, and three variants that greatly reduce the complexities. The pipeline of our approach is demonstrated in Fig. 2.

Naive Non-linear Propagation Model

We design a non-linear propagation model (NLPM) that satisfies the desired requirements of SPN (Liu et al. 2017). Assuming $h^0 \in \mathbb{R}^{m\times n}$ is the initial depth map and $h^N \in \mathbb{R}^{m\times n}$ is the refined depth map after N iterations. $\{h^0, h^N\}$is reshaped as a column-first one-dimensional vector $V^t \in \mathbb{R}^{mn}$, where $t \in \{0, 1, 2 \cdots N\}$. The NLPM could be simplely described as follows,

$$\begin{aligned} V^{t+1} &= GV^t \\ &= \begin{bmatrix} 1-\lambda_{0,0} & w_{0,0}(1,0) & \cdots & w_{0,0}(m,n) \\ w_{1,0}(0,0) & 1-\lambda_{1,0} & \cdots & w_{1,0}(m,n) \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,n}(0,0) & w_{m,n}(1,0) & \cdots & 1-\lambda_{m,n} \end{bmatrix} V^t \end{aligned} \tag{1}$$

where $G \in \mathbb{R}^{mn\times mn}$ is a fixed transformation matrix during the propagation, $w_{i,j}(a,b)$ describes the affinity weight of pixel(i,j) with its neighbors(a,b) and $\lambda_{i,j} = \sum_{a\neq i, b\neq j} w_{i,j}(a,b)$ means that the sum of each row of G is 1.

Ideally, our method adjusts the value of G by applying global attention. Therefore, the NLPM corresponds to the diffusion process with a partial differential equation (PDE) in Eq.(2).

$$\begin{aligned} V^{t+1} &= G^tV^t = (I - D^t + K^t \cdot A)V^t \\ V^{t+1} - V^t &= -(D^t - K^t \cdot A)V^t \\ \partial_t V^{t+1} &= -L^tV^t \end{aligned} \tag{2}$$

Where $K^t \in \mathbb{R}^{mn\times mn}$ is the global attention matrix, D is the diagonal matrix that contains all the $\lambda_{i,j}$, and A is the affinity matrix containing all the $w_{i,j}(a,b)$.

L^t is the Laplace matrix that varies with the number of iterations t, indicating that our model is non-linear. The transformation matrix G^t and the global attention matrix K^t could be concretely written as follows,

$$G^t = \begin{bmatrix} 1-\tilde{\lambda}^t_{0,0} & \tilde{w}^t_{0,0}(1,0) & \cdots & \tilde{w}^t_{0,0}(m,n) \\ \tilde{w}^t_{1,0}(0,0) & 1-\tilde{\lambda}^t_{1,0} & \cdots & \tilde{w}^t_{1,0}(m,n) \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{w}^t_{m,n}(0,0) & \tilde{w}^t_{m,n}(1,0) & \cdots & 1-\tilde{\lambda}^t_{m,n} \end{bmatrix} \tag{3}$$

$$K^t = \begin{bmatrix} 0 & \pi^t_{0,0}(1,0) & \cdots & \pi^t_{0,0}(m,n) \\ \pi^t_{1,0}(0,0) & 0 & \cdots & \pi^t_{1,0}(m,n) \\ \vdots & \vdots & \ddots & \vdots \\ \pi^t_{m,n}(0,0) & \pi^t_{m,n}(1,0) & \cdots & 0 \end{bmatrix} \tag{4}$$

where $\tilde{w}^t_{i,j}(a,b) = \pi^t_{i,j}(a,b)w_{i,j}(a,b)$ is the weight of adaptive affinity, and $\tilde{\lambda}^t_{i,j} = \sum_{a\neq i, b\neq j} \tilde{w}^t_{i,j}(a,b)$.

Diffusion Suppression Operation

The diffusion process would not stop until the end of N iterations, but this is inappropriate for all pixels. If N is too high, the output will be over-smoothing, which is negative for depth completion. To solve this problem, we adopt a diffusion suppression operation that estimates the affinity of refined depth maps in every iteration so that the model converges at an early stage.

We use a diagonal matrix $\bar{D}^t$ to describe the affinity of refined depth maps between t and $t+1$ step in Eq.(5).

$$\bar{D}^t = I - D^t = \begin{bmatrix} 1-\tilde{\lambda}^t_{0,0} & 0 & \cdots & 0 \\ 0 & 1-\tilde{\lambda}^t_{1,0} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1-\tilde{\lambda}^t_{m,n} \end{bmatrix} \tag{5}$$

When the diffusion is completely convergent, the attention matrix K^t is a zero matrix, deducing $\tilde{\lambda}^t_{i,j} = 0$. The diffusion process suppress as follows.

$$V^{t+1} = G^t V^t = \bar{D}^t V^t = I V^t = V^t \tag{6}$$

Neighborhood decoupling of DySPN

Our basic idea of N-step NLPM uses global attention with a large complexity of $O(m^2 n^2 N)$. When we individually assign attention to each sampled neighbor, the complexity drops to $O(k^2 N)$. Meanwhile, the base network needs to estimate N times more feature maps than the linear propagation model, accompanied by a large number of additional calculations.

By analyzing the 7×7 CSPN affinity matrix, we find that the affinity weights of the neighbors at the same distance are correlated. Therefore, we came up with three variants for our DySPN, which decouple the neighborhood into parts regarding different distances. The complexity of our decoupled attention mechanism is only $O(kN)$, which is irrelevant to image size(m, n). Our method is easy to be implemented and lightly embedded in the propagation. As illustrated in Fig. 3, we reduce the amount of neighbors needed while achieving better accuracy in our experiment. Generally, one step propagation of our DySPN could be written as,

$$\begin{aligned} h^{t+1}_{i,j} = & \sum_{k\in\mathbb{Z}_+} \sum_{(a,b)\in N^t_{i,j,k}} \frac{\pi^t_{i,j,k}}{S'_{i,j}} w_{i,j}(a,b) h^t_{a,b} \\ & + \frac{\pi^t_{i,j,0}}{S'_{i,j}} h^t_{i,j} + (1 - \frac{S_{i,j}}{S'_{i,j}}) h^0_{i,j} \end{aligned} \tag{7}$$

where,

$$S_{i,j} = \pi^t_{i,j,0} + \sum_{k\in\mathbb{Z}_+} \sum_{(a,b)\in N^t_{i,j,k}} \pi^t_{i,j,k} w_{i,j}(a,b),$$

$$S'_{i,j} = \pi^t_{i,j,0} + \sum_{k\in\mathbb{Z}_+} \sum_{(a,b)\in N^t_{i,j,k}} \pi^t_{i,j,k} |w_{i,j}(a,b)|.$$

The affinity matrix A is usually extremely sparse because only a limited number of neighbors are sampled. Therefore, we use $N^t_{i,j,k}$ to clearly represent its decoupled neighborhood. The basic form of $N^t_{i,j,k}$ is defined as:

$$\begin{aligned} N^t_{i,j,k} = \{ & (i+p, j+q) | p = \pm k \ or\ q = \pm k, \\ & (-k,-k) \leqslant (p,q) \leqslant (k,k), (p,q) \in \mathbb{R} \}. \end{aligned}$$

where $h^t_{i,j}$ is the pixel(i, j) value of h^t, and $N^t_{i,j,k}$ is the set of neighbors at pixel distance $k \in \mathbb{Z}_+$. $w_{i,j}(a, b)$ is the affinity matrix weight between pixel (i, j) and its neighbour (a, b). Notice that the weight of affinity matrix is normalized by $S_{i,j}$. The spatial and sequential attentions $\pi^t_{i,j,k}$ are activated by a sigmoid function and incorporated into the propagation.Specifically, when the kernel is of size 7, a 7×7 DySPN is shown as the top of Fig. 3.

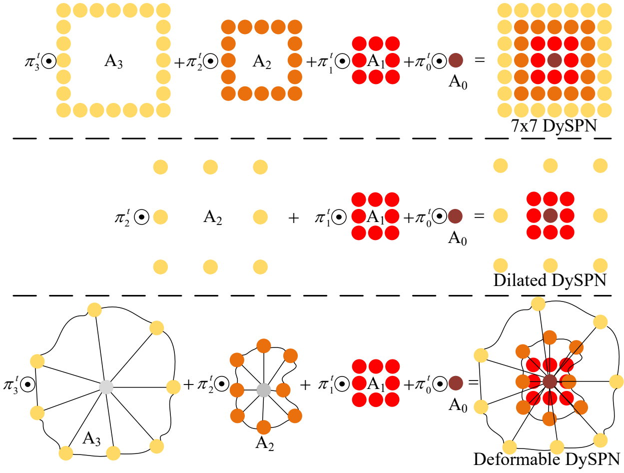

Figure 3: Neighborhood decoupling of three variants. Top: 7×7 DySPN decomposes the neighborhood into four parts based on distance. Middle: Dilated DySPN aggregates two different 3×3 neighborhoods, one of which has a dilation of 3. Bottom: Deformable DySPN uses two deformable 3×3 neighborhoods to obtain different receptive fields.

Dilated DySPN uses fewer neighbors to get a similar receptive field, as shown below:

$$\begin{aligned} N^t_{i,j,k} = \{ & (i+p, j+q) | p \in \{-2k+1, 0, 2k-1\}, \\ & q \in \{-2k+1, 0, 2k-1\}, (p,q) \neq (0,0) \} \end{aligned}$$

At the bottom of Fig. 3, Deformable DySPN has a tradeoff between the number of neighbors and the size of receptive fields. Its neighborhood in pixel(i, j) is defined as follows,

$$\begin{aligned} N^t_{i,j,1} = & \{(i+p, j+q) | p \in \{1,0,-1\}, q \in \{1,0,-1\}, \\ & (p,q) \neq (0,0)\} \\ N^t_{i,j,k} = & \{(i+p, j+q) | (p,q) \in f_{\phi,k}(i,j), (p,q) \in \mathbb{R}, \\ & k \in \{2,3\}\} \end{aligned}$$

where f_ϕ estimates the off-set of neighbors for pixel(i, j).

Network Architecture

An Unet-like (Ronneberger, P.Fischer, and Brox 2015) encoder-decoder network with the backbone of ResNet-34 (He et al. 2016) is proposed as a base network for fair comparison with other SPNs. The head of the base network is modified to fit our DySPN, so that it outputs attention maps π^t_k, an initial depth map h^0, an affinity matrix A, and a confidence prediction of the sparse depth measurement in parallel. The confidence prediction is common in other works such as (Cheng et al. 2020a) (Cheng et al. 2020b) (Hu et al. 2021).

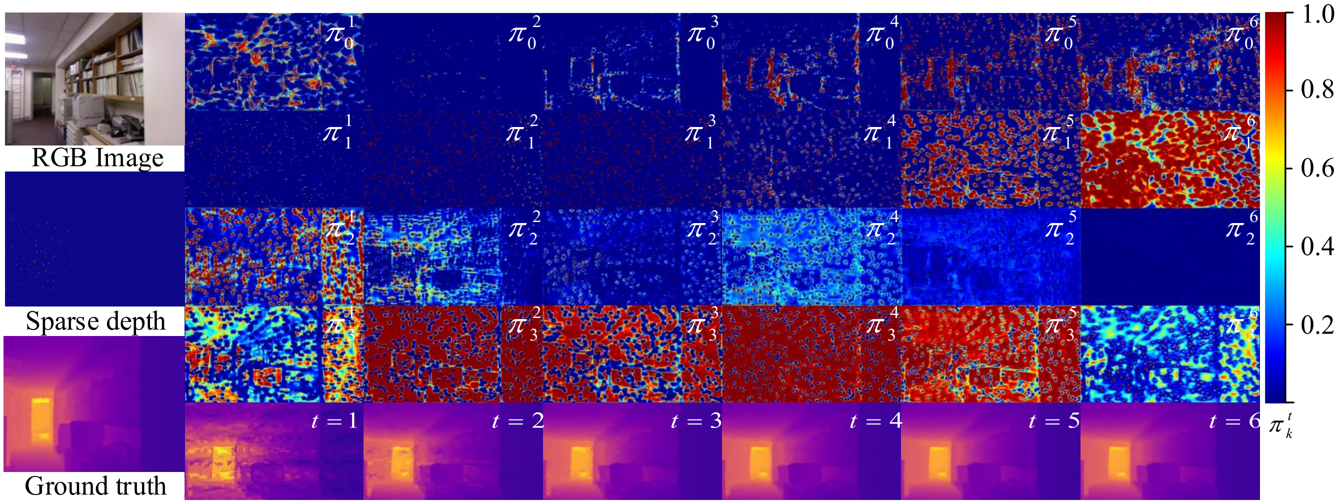

Figure 4: Visualization of attention maps π_k^t activated by sigmoid functions in 7 × 7 DySPN Modules.

Method	RMSE (m)	REL	$\delta_{1.25}$	$\delta_{1.25^2}$	$\delta_{1.25^3}$
CSPN	0.117	0.016	99.2	**99.9**	**100.0**
CSPN++	0.116	-	-	-	-
DeepLiDAR	0.115	0.022	99.3	**99.9**	**100.0**
FCFRNet	0.106	0.015	99.5	**99.9**	**100.0**
ACMNet	0.105	0.015	99.4	**99.9**	**100.0**
GuideNet	0.101	0.015	99.5	**99.9**	**100.0**
TWISE	0.097	0.013	**99.6**	**99.9**	**100.0**
NLSPN	0.092	**0.012**	**99.6**	**99.9**	**100.0**
7×7 DySPN	**0.090**	**0.012**	**99.6**	**99.9**	**100.0**
Dilated -	**0.091**	**0.012**	**99.6**	**99.9**	**100.0**
Deformable -	**0.090**	**0.012**	**99.6**	**99.9**	**100.0**

Table 1: Quantitative evaluation on NYU Depth v2 dataset. The CSPN, CSPN++, DeepLiDAR, FCFRNet, ACMNet, GuideNet, TWISE, NLSPN mean (Ma and Karaman 2018) (Cheng, Wang, and Yang 2018) (Cheng et al. 2020a) (Qiu et al. 2019) (Liu et al. 2021) (Zhao et al. 2021) (Tang et al. 2021) (Imran, Liu, and Morris 2021) (Park et al. 2020), respectively.

Loss Function

L_1 and L_2 loss are used at the same time to improve the performance of DySPN prediction results. Our loss is defined as:

$$
\begin{aligned}
Loss(h^{gt}, h^N) &= \alpha L_1(h^{gt}, h^N) + \beta L_2(h^{gt}, h^N) \\
L_\rho &= \frac{1}{S}\sum_i^m \sum_j^n |(h_{i,j}^N - h_{i,j}^{gt}) \cdot \mathbb{I}(h_{i,j}^{gt} > 0)|^\rho \\
S &= \sum_i^m \sum_j^n \mathbb{I}(h_{i,j}^{gt} > 0)
\end{aligned}
\tag{8}
$$

where h^N is the N-steps refined depth map and h^{gt} is the ground truth; $\mathbb{I}(h_{i,j}^{gt} > 0)$ is an indicator for the validity of h^{gt} at pixel(i,j); $\alpha = \beta = 1$ and $\rho \in \{1, 2\}$.

Experiment

In this section, we follow the same experiment settings with other SPN-based methods (i.e. perform experiments on KITTI DC and NYU Depth v2 dataset) to have a fair comparison. Ablation studies are presented to verify the effectiveness of our method.

Implementation Details

We implement our method on the PyTorch framework and train it with 4 NVIDIA RTX 2080 Ti GPUs. All of our experiments are trained by an ADAM optimizer with β_1=0.9, β_2=0.999. The learning rate starts at 0.001 for the first 30 epochs and reduces to 0.002 with another 10 epochs of training. Furthermore, we use a stochastic depth (Huang et al. 2016) strategy instead of the weight decay to prevent overfitting during training for better performance. For the NYU Depth V2 dataset, the batch size is set to 24 and 500 depth pixels were randomly sampled from the ground truth depth. For the KITTI DC dataset, the batch size is 8. Besides, data augmentation techniques are used, including horizontal random flip and color jitter.

Datasets

NYU Depth V2 dataset (Silberman et al. 2012) is comprised of RGB and depth images captured by a Microsoft Kinect camera in 464 indoor scenes. Following (Cheng, Wang, and Yang 2018) (Park et al. 2020) (Liu et al. 2021), Our model is trained on a subset of 50K images from the official training split and tested on the 654 images from the official labeled test set. Each image is downsized to 320×240, and then 304×228 center-cropping was applied.

KITTI DC dataset (Uhrig et al. 2017) consists of over 90K RGB and LiDAR pairs for training, 1K pairs for validation, and another 1K pairs for testing. The training pairs are

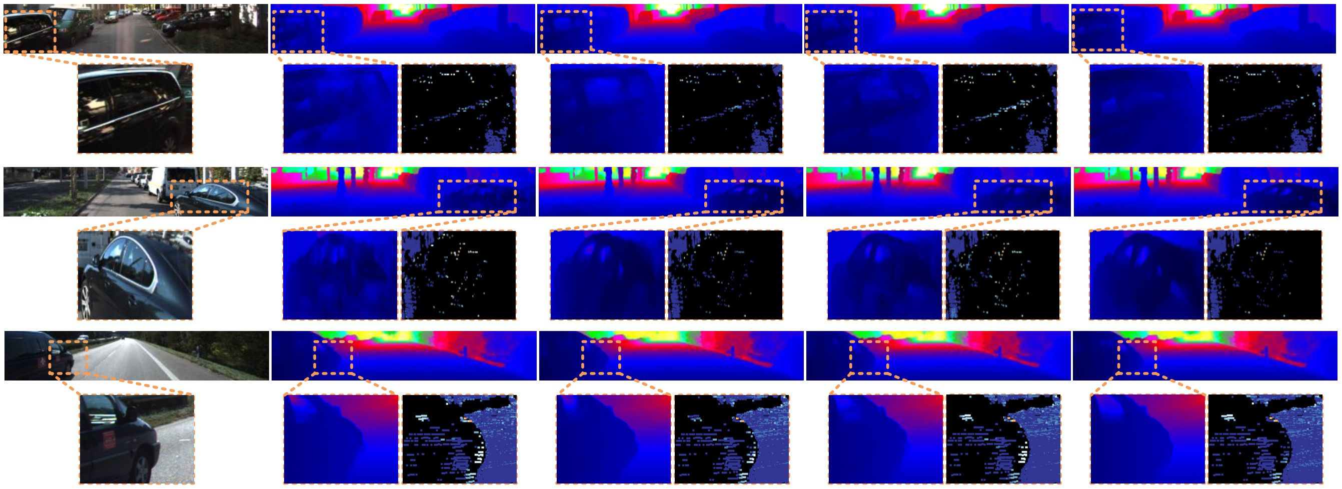
(a)RGB image (b)PENet (c)ACMNet (d)FCFRNet (e)DySPN(Ours)

Figure 5: Qualitative comparisons results with other methods on KITTI DC evaluation. (b) PENet (Hu et al. 2021), (c) ACMNet (Zhao et al. 2021), (d) FCFRNet (Liu et al. 2021).

Method	RMSE (mm)	MAE (mm)	iRMSE (1/km)	iMAE (1/km)
CSPN	1019.64	279.46	2.93	1.15
TWISE	840.20	195.58	2.08	**0.82**
DSPN	766.74	220.36	2.47	1.03
DeepLiDAR	758.38	226.50	2.56	1.15
ACMNet	744.91	206.09	2.08	0.90
CSPN++	743.69	209.28	2.07	0.90
NLSPN	741.68	199.59	1.99	0.84
GuideNet	736.24	218.83	2.25	0.99
FCFRNet	735.81	217.15	2.20	0.98
PENet	730.08	210.55	2.17	0.94
Deformable DySPN	**709.12**	**192.71**	**1.88**	**0.82**

Table 2: Quantitative evaluation on KITTI DC Dataset. The CSPN, TWISE, DSPN, DeepLiDAR, ACMNet, CSPN++, NLSPN, GuideNet, FCFRNet, PENet mean (Cheng, Wang, and Yang 2018) (Imran, Liu, and Morris 2021) (Xu, Yin, and Yao 2020) (Qiu et al. 2019) (Zhao et al. 2021) (Cheng et al. 2020a) (Park et al. 2020) (Tang et al. 2021) (Liu et al. 2021) (Hu et al. 2021), respectively.

top cropped for no LiDAR projection data (i.e., top 100 pixels), and then center cropped to 1216×256. Incidentally, the sparse depth maps are provided by HDL-64 LiDAR contain less than 6% valid values, and the ground truth depth maps are generated by the accumulated LiDAR scans of multiple timestamps with about 14% density.

Metrics

For the NYU Depth V2 dataset, root mean square error (RMSE), mean absolute relative error (REL), and percentage of pixels satisfying δ_τ are selected as the evaluation metrics. For the KITTI DC Dataset, we adopt error metrics same as the dataset benchmark, including RMSE, mean absolute error (MAE), inverse RMSE (iRMSE), and inverse MAE (iMAE). All the metrics for evaluation are shown as follows.

RMSE(mm): $\sqrt{\frac{1}{v}\sum_x(\hat{h}_x - h_x)^2}$

MAE(mm): $\frac{1}{v}\sum_x|\hat{h}_x - h_x|$

iRMSE(1/km): $\sqrt{\frac{1}{v}\sum_x(\frac{1}{\hat{h}_x} - \frac{1}{h_x})^2}$

iMAE(1/km): $\frac{1}{v}\sum_x|\frac{1}{\hat{h}_x} - \frac{1}{h_x}|$

REL: $\frac{1}{v}\sum_x|\frac{\hat{h}_x - h_x}{h_x}|$

δ_τ: $max(\frac{h_x}{\hat{h}_x} - \frac{\hat{h}_x}{h_x}) < \tau, \tau \in \{1.25, 1.25^2, 1.25^3\}$

Evaluation on NYU Depth v2 Dataset

We verified the effectiveness of our method in indoor scenes by evaluating on NYU Depth v2 Dataset. The quantitative comparison results are shown in Table. 1. Our proposed algorithms achieve the best results and outperform other SoTA methods. Fig. 4 visualizes the attention maps learned by our network. Qualitatively, attention maps π_1^t increase significantly with iteration, whereas attention maps π_2^t and π_3^t decrease. These attention maps, such as π_0^3 and π_2^3, contain a lot of boundary information. In contrast with CSPN++ (Cheng et al. 2020a), we find a majority of pixels use a big kernel with a large receptive field at first several iterations for fast recovery, while in the last iteration, pixels around object and surface boundary tend to use a smaller kernel, so that the attention map π_1^t is becoming more and more obvious. We also observe an increasing trend in the attention maps of the diffusion suppression operation(i.e. π_0^t).

Evaluation on KITTI DC Dataset

We also verified the effectiveness of our method in outdoor scenes by evaluating on KITTI DC Dataset. The results of quantitative comparisons are shown in Table. 2. We finally

Method	Iteration	DySPN Module		DS	Training Strategy				Results(Lower the better)	
		Sigmoid	Softmax		Weight Decay			Stochastic Depth	RMSE(mm)	MAE(mm)
					5×10^5	10^5	5×10^6			
7×7 CSPN	12								789.6	201.4
7×7 CSPN	6								811.0	207.3
7×7 DySPN	12	✓							768.1	196.4
-	6	✓							769.0	195.9
-	6		✓						770.0	197.1
-	6	✓		✓					763.9	195.5
-	6	✓		✓	✓				753.2	196.5
-	6	✓		✓		✓			752.9	194.6
-	6	✓		✓			✓		755.8	195.2
-	6	✓		✓		✓		✓	747.0	193.2
-	6	✓		✓				✓	745.8	192.5

Table 3: Ablation study on the KITTI DC validation set. DS means diffusion suppression.

submit the evaluation result of Deformable DySPN, which is the best of our three variations. Our DySPN outperforms all the other works with a significant improvement (nearly 21mm in RMSE) and ranks 1st on KITTI DC evaluation in all metrics (RMSE, MAE, iRMSE, and iMAE) by the time of submission. In Fig. 5, we compare the results of our method with several works. Our work is more consistent in some cases where the depth maps are extremely sparse.

Ablation Studies

In this section, we first perform ablation experiments to verify the effectiveness of our model design. Our experiments are trained on KITTI DC annotated train dataset and tested on the selected validation dataset. A simple 7×7 CSPN (Cheng, Wang, and Yang 2018) is implemented as our baseline model. Table. 3 demonstrates the effectiveness of each design proposed in our 7×7 DySPN, including the DySPN module, the diffusion suppression (DS) operation, and the network training strategies. In addition, we employ a U-net with the backbone of ResNet-34 as our base network.

DySPN module and CSPN module are compared in 6 and 12 iterations respectively, as shown in Table. 3. In particular, the RMSE is decreased from $811.0mm$ of CSPN to $769.0mm$ of DySPN with a significant $42mm$ reduction when the number of iterations is 6. DySPN module also shows better convergence, because the RMSE is not affected when the number of iterations is increased from 6 to 12. We also find that there is little difference between the sigmoid layer and the softmax layer on our module.

Diffusion suppression (DS) operation is easy to be embedded to DySPN module as proposed in Eq. 7. As visualized in Fig. 4, high thermal values gradually appear at the boundary of the attention maps $\{\pi_0^1, \cdots, \pi_0^6\}$, it clearly shows the edge-preserving effect of diffusion suppression. In Table. 3, we observe a reduction in RMSE from $769.0mm$ to $763.9mm$, which means our design is successful.

Training strategy of weight decay is mentioned in many depth completion works (Cheng et al. 2020a) (Hu et al. 2021) (Liu et al. 2021) (Tang et al. 2021) to reduce model over-fitting. In this case, we try different values of weight decay and figure out the best one as shown in Table. 3. In addition, we adopt a stochastic depth strategy(Huang et al. 2016) to play a similar role as weight decay. Experiment results demonstrate that it is better to use stochastic depth alone than with weight decay.

SPN Modules	Iter.	Time (ms)	RMSE (mm)	MAE (mm)
NLSPN	18	55	-	-
PENet(C2)	12	15	757.2	209.0
7×7 CSPN	12	70	789.6	201.4
7×7 DySPN	6	38	745.8	192.5
Dilated -	6	6.9	748.5	193.8
Deformable -	6	10.3	739.4	191.4

Table 4: Runtime and preformance of three variants of DySPN on the KITTI DC validation set.

Dilated DySPN and Deformable DySPN improve the effect and speed with fewer iterations and fewer neighbor samples, as illustrated in Table. 4. Specifically, Dilated DySPN module is designed for a better speed-accuracy trade-off. It shows not only over 2× faster than the C2 module of PENet but also improves RMSE from $757.2mm$ to $748.5mm$. Deformable DySPN is improved on precision. As summarized in Table. 2, it ranks 1st on the KITTI DC benchmark at the time of submission.

Conclusion

In this paper, we proposed DySPN with a non-linear propagation model, which applies spatial and sequential attention to generate a series of adaptive affinity matrices. Furthermore, we presented a diffusion suppression operation to avoid the over-smoothing problem. To reduce complexity, we elaborated on three variants, including 7×7 DySPN, Dilated DySPN, and Deformable DySPN, which have different speed-accuracy trade-offs. Our ablation studies verified the superior performance of our DySPN variations on both indoor and outdoor depth completion datasets. By the way, we believe that DySPN can not only be used for image-guided depth completion but also be applied to other aspects, such as image segmentation and cellular automata modeling.

References

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 39(12): 2481–2495.

Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; and Liu, Z. 2020. Dynamic Convolution: Attention Over Convolution Kernels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11027–11036.

Cheng, X.; Wang, P.; Chenye, G.; and Yang, R. 2020a. CSPN++: Learning Context and Resource Aware Convolutional Spatial Propagation Networks for Depth Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 10615–10622.

Cheng, X.; Wang, P.; and Yang, R. 2018. Depth Estimation via Affinity Learned with Convolutional Spatial Propagation Network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 108–125.

Cheng, X.; Wang, P.; and Yang, R. 2020. Learning Depth with Convolutional Spatial Propagation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(10): 2361–2379.

Cheng, X.; Wang, P.; Zhou, Y.; Guan, C.; and Yang, R. 2020b. Omnidirectional Depth Extension Networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 589–595.

Chiu, Y.-P.; Leou, J.-J.; and Hsiao, H.-H. 2014. Super-resolution reconstruction for Kinect 3D data. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2712–2715.

Eldesokey, A.; Felsberg, M.; and Khan, F. S. 2020. Confidence Propagation through CNNs for Guided Sparse Depth Regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(10): 2423–2436.

Hawe, S.; Kleinsteuber, M.; and Diepold, K. 2011. Dense disparity maps from sparse disparity measurements. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2126–2133.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hu, H.-N.; Cai, Q.-Z.; Wang, D.; Lin, J.; Sun, M.; Krahenbuhl, P.; Darrell, T.; and Yu, F. 2019. Joint Monocular 3D Vehicle Detection and Tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5390–5399.

Hu, M.; Wang, S.; Li, B.; Ning, S.; Fan, L.; and Gong, X. 2021. PENet: Towards Precise and Efficient Image Guided Depth Completion. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K. Q. 2016. Deep Networks with Stochastic Depth. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 646–661.

Imran, S.; Liu, X.; and Morris, D. 2021. Depth Completion with Twin Surface Extrapolation at Occlusion Boundaries. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2583–2592.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic Filter Networks. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems (NIPS)*, volume 29. Curran Associates, Inc.

Ku, J.; Harakeh, A.; and Waslander, S. L. 2018. In Defense of Classical Image Processing: Fast Depth Completion on the CPU. In *15th Conference on Computer and Robot Vision (CRV)*, 16–22.

Liu, L.; Song, X.; Lyu, X.; Diao, J.; Wang, M.; Liu, Y.; and Zhang, L. 2021. FCFR-Net: Feature Fusion based Coarse-to-Fine Residual Learning for Depth Completion. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 2136–2144.

Liu, S.; De Mello, S.; Gu, J.; Zhong, G.; Yang, M.-H.; and Kautz, J. 2017. Learning Affinity via Spatial Propagation Networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 30.

Ma, F.; and Karaman, S. 2018. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In *IEEE International Conference on Robotics and Automation (ICRA)*, 4796–4803.

Ma, N.; Zhang, X.; Huang, J.; and Sun, J. 2020a. WeightNet: Revisiting the Design Space of Weight Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 776–792.

Ma, X.; Liu, S.; Xia, Z.; Zhang, H.; Zeng, X.; and Ouyang, W. 2020b. Rethinking Pseudo-LiDAR Representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 311–327.

Park, J.; Joo, K.; Hu, Z.; Liu, C.-K.; and So Kweon, I. 2020. Non-local Spatial Propagation Network for Depth Completion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 120–136.

Park, J.; Kim, H.; Tai, Y.-W.; Brown, M. S.; and Kweon, I. 2011. High quality depth map upsampling for 3D-TOF cameras. In *International Conference on Computer Vision (ICCV)*, 1623–1630.

Qiu, J.; Cui, Z.; Zhang, Y.; Zhang, X.; Liu, S.; Zeng, B.; and Pollefeys, M. 2019. DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene From Sparse LiDAR Data and Single Color Image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3308–3317.

Ronneberger, O.; P.Fischer; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, 234–241. Springer. (available on arXiv:1505.04597 [cs.CV]).

Silberman, N.; Hoiem, D.; Kohli, P.; and Fergus, R. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 746–760.

Su, H.; Jampani, V.; Sun, D.; Gallo, O.; Learned-Miller, E.; and Kautz, J. 2019. Pixel-Adaptive Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11158–11167.

Tang, J.; Tian, F.-P.; Feng, W.; Li, J.; and Tan, P. 2021. Learning Guided Convolutional Network for Depth Completion. *IEEE Transactions on Image Processing (TIP)*, 30: 1116–1129.

Uhrig, J.; Schneider, N.; Schneider, L.; Franke, U.; Brox, T.; and Geiger, A. 2017. Sparsity Invariant CNNs. In *International Conference on 3D Vision (3DV)*, 11–20.

Van Gansbeke, W.; Neven, D.; De Brabandere, B.; and Van Gool, L. 2019. Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty. In *16th International Conference on Machine Vision Applications (MVA)*, 1–6.

Wang, C.; Xu, D.; Zhu, Y.; Martín-Martín, R.; Lu, C.; Fei-Fei, L.; and Savarese, S. 2019. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3338–3347.

Weickert, J. 1998. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart.

Wong, A.; and Soatto, S. 2021. Unsupervised Depth Completion with Calibrated Backprojection Layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 12747–12756.

Xu, H.; and Saenko, K. 2016. Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Xu, Z.; Yin, H.; and Yao, J. 2020. Deformable Spatial Propagation Networks For Depth Completion. In *IEEE International Conference on Image Processing (ICIP)*, 913–917.

Yang, B.; Bender, G.; Le, Q. V.; and Ngiam, J. 2019. CondConv: Conditionally Parameterized Convolutions for Efficient Inference. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32. Curran Associates, Inc.

Yang, N.-E.; Kim, Y.-G.; and Park, R.-H. 2012. Depth hole filling using the depth distribution of neighboring regions of depth holes in the Kinect sensor. In *IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, 658–661.

Yang, Q.; Yang, R.; Davis, J.; and Nister, D. 2007. Spatial-Depth Super Resolution for Range Images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8.

Zhang, J.; and Singh, S. 2015. Visual-lidar odometry and mapping: low-drift, robust, and fast. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2174–2181.

Zhang, Y.; Zhang, J.; Wang, Q.; and Zhong, Z. 2020. DyNet: Dynamic Convolution for Accelerating Convolutional Neural Networks. arXiv:2004.10694.

Zhao, S.; Gong, M.; Fu, H.; and Tao, D. 2021. Adaptive Context-Aware Multi-Modal Network for Depth Completion. *IEEE Transactions on Image Processing (TIP)*, 30: 5264–5276.

Zhou, J.; Jampani, V.; Pi, Z.; Liu, Q.; and Yang, M.-H. 2021. Decoupled Dynamic Filter Networks. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, S.; Zhang, R.; Zhou, L.; Shen, T.; Fang, T.; Tan, P.; and Quan, L. 2018. Very Large-Scale Global SfM by Distributed Motion Averaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4568–4577.