Non-Local Spatial Propagation Network for Depth Completion

Jinsun Park[1], Kyungdon Joo[2], Zhe Hu[3], Chi-Kuei Liu[3], and In So Kweon[1]

[1] Korea Advanced Institute of Science and Technology, Republic of Korea
{zzangjinsun, iskweon77}@kaist.ac.kr
[2] Robotics Institute, Carnegie Mellon University
kjoo@andrew.cmu.edu
[3] Hikvision Research America

Abstract. In this paper, we propose a robust and efficient end-to-end non-local spatial propagation network for depth completion. The proposed network takes RGB and sparse depth images as inputs and estimates non-local neighbors and their affinities of each pixel, as well as an initial depth map with pixel-wise confidences. The initial depth prediction is then iteratively refined by its confidence and non-local spatial propagation procedure based on the predicted non-local neighbors and corresponding affinities. Unlike previous algorithms that utilize fixed-local neighbors, the proposed algorithm effectively avoids irrelevant local neighbors and concentrates on relevant non-local neighbors during propagation. In addition, we introduce a learnable affinity normalization to better learn the affinity combinations compared to conventional methods. The proposed algorithm is inherently robust to the mixed-depth problem on depth boundaries, which is one of the major issues for existing depth estimation/completion algorithms. Experimental results on indoor and outdoor datasets demonstrate that the proposed algorithm is superior to conventional algorithms in terms of depth completion accuracy and robustness to the mixed-depth problem. Our implementation is publicly available on the project page.[4]

Keywords: Depth completion, Non-local, Spatial propagation network

1 Introduction

Depth estimation has become an important problem in recent years with the rapid growth of computer vision applications, such as augmented reality, unmanned aerial vehicle control, autonomous driving, and motion planning. To obtain a reliable depth prediction, information from various sensors is utilized, *e.g.*, RGB cameras, radar, LiDAR, and ultrasonic sensors [2, 3]. Depth sensors, such as LiDAR sensors, produce accurate depth measurements with high frequency. However, the density of the acquired depth is often sparse due to hardware limitations, such as the number of scanning channels. To overcome such

---

[4] https://github.com/zzangjinsun/NLSPN_ECCV20
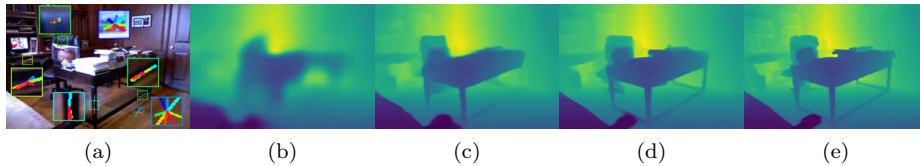
(a)   (b)   (c)   (d)   (e)

Fig. 1. Example of the depth completion on the NYU Depth V2 dataset [29].
(a) RGB image and a few samples of the estimated non-local neighbors. Depth completion results by (b) direct regression [21], (c) local propagation [9], and (d) non-local propagation (ours), respectively, and (e) the ground truth.

limitations, there have been a lot of works to estimate dense depth information based on the given sparse depth values, called *depth completion*.

Early methods for depth completion [30, 10] rely only on sparse measurement. Therefore, their predictions suffer from unwanted artifacts, such as blurry and mixed-depth values (*i.e.*, mixed-depth problem). Because RGB images show subtle changes of color and texture, recent methods use RGB images as the guidance to predict accurate dense depth maps.

Direct depth completion algorithms [30, 21] take RGB or RGB-D images and directly infer a dense depth using a deep convolutional neural network (CNN). These direct algorithms have shown superior performance compared to conventional ones; however, they still generate blurry depth maps near depth boundaries. Soon after, this phenomenon is alleviated by recent affinity-based spatial propagation methods [9, 32]. By learning affinities for local neighbors and iteratively refining depth predictions, the final dense depth becomes more accurate. Nonetheless, previous propagation networks [19, 9] have an explicit limitation that they have a fixed-local neighborhood configuration for propagation. Fixed-local neighbors often have irrelevant information that should not be mixed with reference information, especially on depth boundaries. Hence, they still suffer from the mixed-depth problem in the depth completion task (see Fig. 1(c)).

To tackle the problem, we propose a Non-Local Spatial Propagation Network (NLSPN) that predicts non-local neighbors for each pixel (*i.e.*, where the information should come from) and then aggregates relevant information using the spatially-varying affinities (*i.e.*, how much information should be propagated), which are also predicted from the network. By relaxing the fixed-local neighborhood configuration, the proposed network can avoid irrelevant local neighbors affiliated with other adjacent objects. Therefore, our method is inherently robust to the mixed-depth problem. In addition, based on our analysis of conventional affinity normalization schemes, we propose a learnable affinity normalization method that has a larger representation capability of affinity combinations. It enables more accurate affinity estimation and thus improves the propagation among non-local neighbors. To further improve robustness to outliers from input and inaccurate initial prediction, we predict the confidence of the initial dense depth simultaneously, and it is incorporated into the affinity normalization to minimize the propagation of unreliable depth values. Experimental results on the indoor [29] and outdoor [30] datasets demonstrate that our method achieves superior depth completion performance compared with state-of-the-art methods.

2   Related Work

Depth Estimation and Completion The objective of depth estimation is to generate dense depth predictions based on various input information, such as a single RGB image, multi-view images, sparse LiDAR measurements, and so on. Conventional depth estimation algorithms often utilize information from a single modality. Eigen *et al.* [11] used a multi-scale neural network to predict depth from a single image. In the method introduced by Zbontar and LeCun [35], the deep features of image patches are extracted from stereo rectified images, and then the disparity is determined by searching for the most similar patch along the epipolar line. Depth estimation with accurate but sparse depth information (*i.e.*, depth completion) has been intensively explored as well. Uhrig *et al.* [30] proposed sparsity invariant CNNs to predict a dense depth map given a sparse depth image from a LiDAR sensor. Ma and Sertac [21] introduced a method to construct a 4D volume by concatenating RGB and sparse depth images and then feed it into an encoder-decoder CNN for the final prediction. Chen *et al.* [7] adopted a fusion of 2D convolution and 3D continuous convolution to effectively consider the geometric configuration of 3D points.

Spatial Propagation Network Although direct depth completion algorithms have demonstrated decent performance, sparse-to-dense propagation with accurate guidance from different modalities (*e.g.*, an RGB image) is a more effective way to obtain dense prediction from sparse inputs [9, 32, 17, 22]. Liu *et al.* [19] proposed a spatial propagation network (SPN) to learn local affinities. The SPN learns task-specific affinity values from large-scale data, and it can be applied to a variety of high-level vision tasks, including depth completion and semantic segmentation. However, the individual three-way connection in four-direction is adopted for spatial propagation, which is not suitable for considering all local neighbors simultaneously. This limitation was overcome by Cheng *et al.* [9], who proposed a convolutional spatial propagation network (CSPN) to predict affinity values for local neighbors and update all the pixels simultaneously with their local context for efficiency. However, both the SPN and the CSPN rely on fixed-local neighbors, which could be from irrelevant objects. Therefore, the propagation based on those neighbors would result in mixed-depth values, and the iterative propagation procedure used in their architectures would increase the impact. Moreover, the fixed neighborhood patterns restrict the usage of relevant but wide-range (*i.e.*, non-local) context within the image.

Non-Local Network The importance of non-local information has been widely explored in various vision tasks [5, 31, 34, 28]. Recently, a non-local block in deep neural networks was proposed by Wang *et al.* [31]. It consists of pairwise affinity calculation and feature-processing modules. The authors demonstrated the effectiveness of non-local blocks by embedding them into existing deep networks for video classification and image recognition. These methods showed significant improvement over local methods.

Our Work Unlike previous algorithms [19, 9, 32], our network is trained to predict non-local neighbors with corresponding affinities. In addition, our learnable affinity normalization algorithm searches for the optimal affinity space, which has
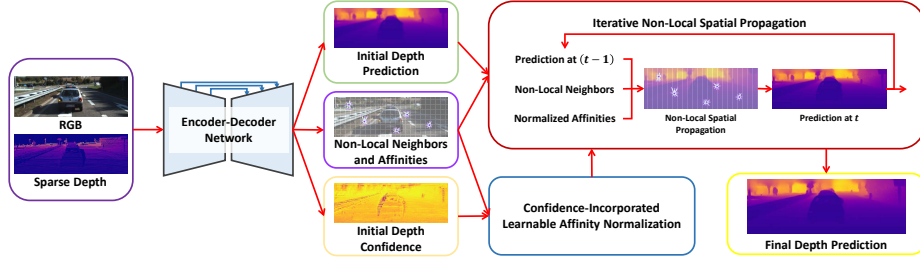
Fig. 2. Overview of the proposed algorithm. The encoder-decoder network is built upon the residual network [13]. Given RGB and sparse depth images, an initial dense depth and its confidence, non-local neighbors, and corresponding affinities are predicted from the network. Then non-local spatial propagation is conducted iteratively with the confidence-incorporated learnable affinity normalization.

not been explored in conventional algorithms [6, 19, 9]. Furthermore, we incorporate the confidence of the initial dense depth prediction (which will be refined by propagation procedure) into affinity normalization to minimize the propagation of unconfident depth values. Figure 2 shows an overview of our algorithm. Each component will be described in subsequent sections in detail.

3   Non-Local Spatial Propagation

The goal of spatial propagation is to estimate missing values and refine less confident values by propagating neighbor observations with corresponding affinities (*i.e.*, similarities). Spatial propagation has been utilized as one of the key modules in various computer vision applications [24, 17, 16]. In particular, spatial propagation is suitable for the depth completion task [19, 9, 32], and its superior performance compared to direct regression algorithms has been demonstrated [30, 21]. In this section, we first briefly review the local SPNs and their limitations, and then describe the proposed non-local SPN.

3.1   Local Spatial Propagation Network

Let $\mathbf{X} = (x_{m,n}) \in \mathbb{R}^{M \times N}$ denote a 2D map to be updated by spatial propagation, where $x_{m,n}$ denotes the pixel value at (m,n). The propagation of $x_{m,n}$ at the step t with its local neighbors, denoted by $\mathcal{N}_{m,n}$, is defined as follows:

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

where (m,n) and (i,j) are the coordinates of reference and neighbor pixels, respectively; $w_{m,n}^c$ represents the affinity of the reference pixel; and $w_{m,n}^{i,j}$ indicates the affinity between the pixels at (m,n) and (i,j). The first term in the right-hand side represents the propagation of the reference pixel, while the second term stands for the propagation of its neighbors weighted by the corresponding
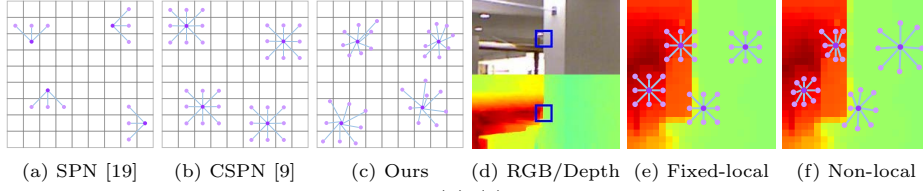
(a) SPN [19]   (b) CSPN [9]   (c) Ours   (d) RGB/Depth (e) Fixed-local   (f) Non-local

Fig. 3. Visual comparison of SPNs. (a)-(c) Examples of neighbor configurations of the (a) SPN [19], (b) CSPN [9], and (c) NLSPN (ours), where purple and light purple pixels denote reference and neighboring pixels, respectively. Compared to the others, our neighbor configuration is highly flexible, and can be fractional. (d)-(f) Comparison of fixed-local and non-local configurations for various situations. The fixed-local configuration (e) cannot utilize relevant information beyond the fixed-local region. In contrast, the non-local configuration (f) avoids this problem effectively by predicting and utilizing relevant neighbors at various distances without limitation.

affinities. The affinity of the reference pixel $w_{m,n}^c$ (*i.e.*, how much the original value will be preserved) is obtained as

$$w_{m,n}^c = 1 - \sum_{(i,j) \in \mathcal{N}_{m,n}} w_{m,n}^{i,j}. \tag{2}$$

Spatial Propagation Network The original SPN [19] is formulated on the configuration of three-way local connections, where each pixel is linked to three adjacent pixels from the previous row or column (see Fig. 3(a)). For instance, the local neighbors of the pixel at (m, n) for top-to-bottom propagation (*i.e.*, vertical) in the SPN, denoted by $\mathcal{N}_{m,n}^S$, are defined as follows:

$$\mathcal{N}_{m,n}^S = \{x_{m+p,n+q} \mid p = -1, q \in \{-1, 0, 1\}\}. \tag{3}$$

The local neighbors for other directions (*i.e.*, bottom-to-top, left-to-right and right-to-left) can be defined in similar ways. Figure 3(a) shows several examples of $\mathcal{N}^S$ for other directions. Note that the SPN updates rows or columns in **X** sequentially. Thus, a natural limitation of the three-way connection is that it does not explore information from all the directions simultaneously.

Convolutional Spatial Propagation Network To consider all the possible propagation directions together, the original SPN propagates in four directions individually. Then it utilizes max-pooling to integrate those predictions [19]. The CSPN [9] addresses the inefficiency issue by simplifying separate propagations via convolution operation at each propagation step. For the CSPN with a 3×3 local window size, the local neighbors $\mathcal{N}_{m,n}^{CS}$ are defined as follows:

$$\mathcal{N}_{m,n}^{CS} = \{x_{m+p,n+q} \mid p \in \{-1, 0, 1\}, q \in \{-1, 0, 1\}, (p, q) \neq (0, 0)\}. \tag{4}$$

Figure 3(b) shows some examples of $\mathcal{N}^{CS}$. For more details of each network (the SPN and the CSPN), please refer to earlier works [19, 9].

3.2 Non-Local Spatial Propagation Network

The SPN and the CSPN are effective in propagating information from more confident areas into less confident ones with data-dependent affinities. However,

their potential improvement is inherently limited by the fixed-local neighborhood configuration (Fig. 3(e)). The fixed-local neighborhood configuration ignores object/depth distribution within the local area; thus, it often results in mixed-depth values of foreground and background objects after propagation. Although affinities predicted from the network can alleviate the depth mixing between irrelevant pixels to a certain degree, they can hardly avoid incorrect predictions and hold up the use of appropriate neighbors beyond the local area.

To resolve the above issues, we introduce a deep neural network that estimates the neighbors of each pixel beyond the local region (*i.e.*, non-local) based on color and depth information within a wide area. The non-local neighbors $\mathcal{N}_{m,n}^{\mathrm{NL}}$ are defined as follows:

$$\mathcal{N}_{m,n}^{\mathrm{NL}} = \{x_{m+p,n+q} \mid (p,q) \in f_{\phi}(\mathbf{I}, \mathbf{D}, m, n),\ p, q \in \mathbb{R}\}, \tag{5}$$

where $\mathbf{I}$ and $\mathbf{D}$ are the RGB and sparse depth images, respectively, and $f_{\phi}(\cdot)$ is the non-local neighbor prediction network that estimates K neighbors for each pixel, under the learnable parameters ϕ. We adopt an encoder-decoder CNN architecture for $f_{\phi}(\cdot)$, which will be described in Sec. 5.1. It should be noted that p and q are real numbers in Eq. (5); thus, the non-local neighbors can be defined to sub-pixel accuracy, as illustrated in Fig. 3(c).

Figure 3(f) shows some examples of appropriate and desired non-local neighbors near depth boundaries. In the fixed-local setup, affinity learning learns how to encourage the influence of the related pixels and suppress that of unrelated ones simultaneously. On the contrary, affinity learning with the non-local setup concentrates on relevant neighbors, and this facilitates the learning process.

4   Confidence-Incorporated Affinity Learning

Affinity learning is one of the key components in SPNs, which enables accurate and stable propagation. Conventional affinity-based algorithms utilize color statistics or hand-crafted features [17, 27, 16]. Recent affinity learning methods [18, 19, 9] adopt deep neural networks to predict affinities and show substantial performance improvement. In these methods, affinity normalization plays an important role to stabilize the propagation process.

In this section, we analyze the conventional normalization approach and its limitation, and then propose a normalization approach in a learnable way. Moreover, we incorporate the confidence of the initial prediction during normalization to suppress negative effects from unreliable depth values during propagation.

4.1   Affinity Normalization

The purpose of affinity normalization is to ensure stability during propagation. For stability, the norm of the temporal Jacobian of x, $\partial x^t / \partial x^{t-1}$ should be equal to or less than one [19]. Under the spatial propagation formulation in Eq. (1), this condition would be satisfied if $\sum_{(i,j) \in \mathcal{N}_{m,n}} |w_{m,n}^{i,j}| \leq 1,\ \forall m, n$. To enforce
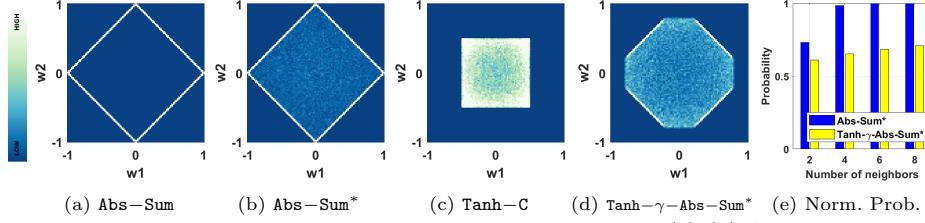
(a) Abs−Sum   (b) Abs−Sum* (c) Tanh−C   (d) Tanh−γ−Abs−Sum* (e) Norm. Prob.

Fig. 4. Illustration of affinity normalization schemes. (a)-(d) Affinity distribution after various normalization schemes for the 2-neighbor case. Color bar is shown on the left. (e) Probabilities of normalization with different strategies for each number of neighbors. Please refer to the text for details.

the condition, previous works [19, 9] normalize affinities by the absolute-sum (dubbed Abs−Sum) as follows:

$$w_{m,n}^{i,j} = \hat{w}_{m,n}^{i,j} / \sum_{(i,j)\in\mathcal{N}_{m,n}} |\hat{w}_{m,n}^{i,j}|, \qquad (6)$$

where $\hat{w}$ denotes the raw affinity before normalization. Although the stability condition is satisfied by Abs−Sum, it has a problem in that the viable combinations of normalized affinities are biased to a narrow high-dimensional space.

Without loss of generality, we first analyze the biased affinity problem using a toy example of the 2-neighbor case and then present solutions to the issue. In the 2-neighbor case, we denote affinities of the two neighbors as w_1 and w_2 with a slight abuse of notation. We assume that the unnormalized affinities are sampled from the standard normal distribution, $\mathsf{N}(0,1)$ for simplicity.

For the Abs−Sum, the normalized affinities lie on the lines satisfying $|w_1| + |w_2| = 1$ (referred to as A_1), as shown in Fig. 4(a). This limits the usage of potentially advantageous affinity configuration within the area $|w_1| + |w_2| < 1$ (referred to as A_2). To fully explore the affinity configuration $|w_1| + |w_2| \leq 1$, a simple remedy is to apply Eq. (6) only when $\sum_i |w_i| > 1$ (noted as Abs−Sum*). Figure 4(b) shows the affinity distribution of our simple remedy. However, the affinities normalized by Abs−Sum* still have a high chance to fall on A_1. Indeed, with the increasing number of neighbors K, the affinities are more likely to lie on A_1. (*e.g.*, the normalization probability is 0.985 when $K = 4$). Figure 4(e) (blue bars) shows the probability of affinities falling on A_1 with various K values.

One way to reduce the bias is to limit the range of raw affinities [20], for example, to $[-1/C, 1/C]$ using the hyperbolic tangent function ($\tanh(\cdot)$) with a normalization factor C. We refer to this normalization procedure as Tanh−C, which is defined as follows:

$$w_{m,n}^{i,j} = \tanh(\hat{w}_{m,n}^{i,j})/C, \qquad C \geq K, \qquad (7)$$

where the condition $C \geq K$ enforces the normalized affinities to guarantee $\sum_{(i,j)\in\mathcal{N}_{m,n}} |w_{m,n}^{i,j}| \leq 1$; therefore, this condition ensures stability. Figure 4(c) shows the affinity distribution of Tanh−C when $C = 2$. With a sacrifice of boundary values, Tanh−C enables a more balanced affinity distribution. Moreover, the
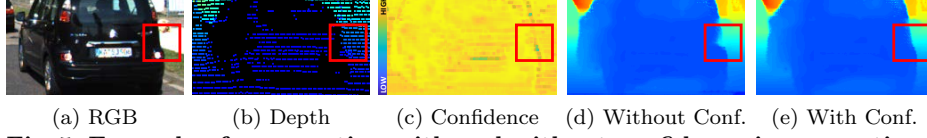
(a) RGB   (b) Depth   (c) Confidence   (d) Without Conf.   (e) With Conf.

Fig. 5. Example of propagation with and without confidence incorporation.

optimal value of C in $\mathtt{Tanh-C}$ may vary depending on the training task, *e.g.*, the number of neighbors, the activation functions, and the dataset.

To determine the optimal value for the task, we propose to learn the normalization factor together with non-local affinities, and apply the normalization only when $\sum_{(i,j)\in\mathcal{N}_{m,n}}|w_{m,n}^{i,j}| > 1$. The affinity of the proposed normalization, referred to as $\mathtt{Tanh-\gamma-Abs-Sum^*}$, is defined as follows:

$$w_{m,n}^{i,j} = \mathtt{tanh}(\hat{w}_{m,n}^{i,j})/\gamma, \quad \gamma_{min} \leq \gamma \leq \gamma_{max}, \tag{8}$$

where γ denotes the learnable normalization parameter, and γ_{min} and γ_{max} are the minimum and maximum values that can be empirically set. Figure 4(d) shows an example of $\mathtt{Tanh-\gamma-Abs-Sum^*}$ when $\gamma = 1.25$. Here, $\mathtt{Tanh-\gamma-Abs-Sum^*}$ can be viewed as a mixture of $\mathtt{Abs-Sum^*}$ and $\mathtt{Tanh-C}$ (see Figs. 4(b) and (c)). The probability of affinities falling on the boundary with respect to the number of neighbors with $\gamma = K/2$ is shown in Fig. 4(e) (yellow bars). Compared to $\mathtt{Abs-Sum^*}$, $\mathtt{Tanh-\gamma-Abs-Sum^*}$ still has a chance to avoid normalization, and it allows us to explore more diverse affinities with a larger number of neighbors.

4.2   Confidence-Incorporated Affinity Normalization

In the existing propagation frameworks [17, 27, 16, 18, 19, 9], the affinity depicts the correlation between pixels and provides guidance for propagation based on similarity. In this case, each pixel in the map is treated equally without consideration of its reliability. However, in the depth completion task, different pixels should be weighted based on their reliability. For example, information from unreliable pixels (*e.g.*, noisy pixels and pixels on depth boundaries) should not be propagated into neighbors regardless of their affinity to the neighboring pixels. The recent work DepthNormal [32] addresses this problem with confidence prediction. It utilizes confidence as a mask for the weighted summation of input and prediction for seed point preservation. However, it does not fully prevent the propagation of incorrect depth values because weighted summation is conducted before each propagation separately.

In this work, we consider the confidence map of pixels and combine it with affinity normalization. That is, we predict not only the initial dense depth but also its confidence, and then the confidence is incorporated into affinity normalization to reduce disturbances from unreliable depths during propagation. The affinity of the confidence-incorporated $\mathtt{Tanh-\gamma-Abs-Sum^*}$ is defined as follows:

$$w_{m,n}^{i,j} = c^{i,j} \cdot \mathtt{tanh}(\hat{w}_{m,n}^{i,j})/\gamma, \tag{9}$$

where $c^{i,j} \in [0,1]$ denotes the confidence of the pixel at (i,j).

Figure 5(d) shows an example of a confidence-agnostic depth estimation result. Some noisy input depth points generate unreliable depth values with low confidences (see Fig. 5(c)). Without using confidence, the noisy and less confident pixels would harm their neighbor pixels during propagation and lead to unpleasing artifacts (see Fig. 5(d)). After the incorporation of confidence into normalization, our algorithm can successfully eliminate the impact of unconfident pixels and generate more accurate depth estimation, as shown in Fig. 5(e).

5   Depth Completion Network

In this section, we describe network architecture and loss functions for network training. The proposed NLSPN mainly consists of two parts: (1) an encoder-decoder architecture for the initial depth map, a confidence map and non-local neighbors prediction with their raw affinities, and (2) a non-local spatial propagation layer with a learnable affinity normalization.

5.1   Network Architecture

The encoder-decoder part of the proposed network is built upon residual networks [13], and it extracts high-level features from RGB and sparse depth images. Additionally, we adopt the encoder-decoder feature connection strategy [26, 9] to simultaneously utilize low-level and high-level features.

In Fig. 2, we provide an overview of our algorithm. Features from the encoder-decoder network are shared for the initial dense depth, confidence, non-local neighbor, and raw affinity estimation. Then non-local spatial propagation is conducted in an iterative manner. As described in Sec. 3.2, non-local neighbors can have fractional coordinates. To better incorporate fractional coordinates into training, differentiable sampling [15, 36] is adopted during propagation. We note that our non-local propagation can be efficiently calculated by deformable convolutions [36]. Therefore, each propagation requires a simple forward step of deformable convolution with our affinity normalization. Please refer to the supplementary material for the detailed network configuration.

5.2   Loss Function

For accurate prediction of the dense depth map, we train our network with ℓ_1 or ℓ_2 loss as a reconstruction loss with the ground truth depth as follows:

$$L_{recon}(\mathbf{D}^{gt}, \mathbf{D}^{pred}) = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \big| \, d_v^{gt} - d_v^{pred} \, \big|^{\rho}, \tag{10}$$

where $\mathbf{D}^{gt}$ is the ground truth depth; $\mathbf{D}^{pred}$ is the prediction from our algorithm; and d_v, $\mathcal{V}$, and $|\mathcal{V}|$ denote the depth values at pixel index v, valid pixels of $\mathbf{D}^{gt}$, and the number of valid pixels, respectively. Here, ρ is set to 1 for ℓ_1 loss and 2 for ℓ_2 loss. Note that we do not have any supervision on the confidence because there is no ground truth; therefore, it is indirectly trained based on L_{recon}.
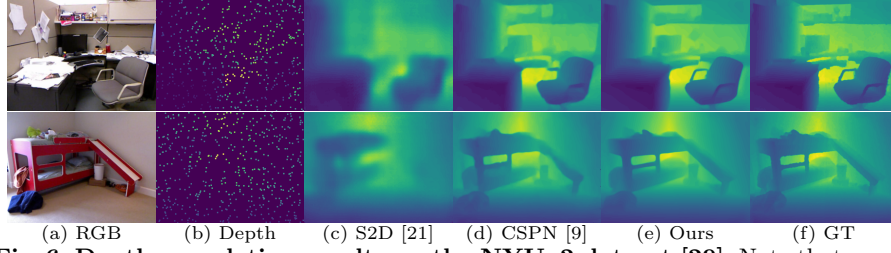
(a) RGB   (b) Depth   (c) S2D [21]   (d) CSPN [9]   (e) Ours   (f) GT

Fig. 6. Depth completion results on the NYUv2 dataset [29]. Note that sparse depth images are dilated for visualization.

6 Experimental Results

In this section, we first describe implementation details and the training environment. After that, quantitative and qualitative comparisons to previous algorithms on indoor and outdoor datasets are presented. We also present ablation studies to verify the effectiveness of each component of the proposed algorithm.

The proposed method was implemented using PyTorch [23] with NVIDIA Apex [1] and trained with a machine equipped with Intel Xeon E5-2620 and 4 NVIDIA GTX 1080 Ti GPUs. For all our experiments, we adopted an ADAM optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the initial learning rate of 0.001. The network training took about 1 and 3 days on the NYU Depth V2 [29] and KITTI Depth Completion [30] datasets, respectively. We adopted the ResNet34 [13] as our encoder-decoder baseline network. The number of non-local neighbors was set to 8 for a fair comparison to other algorithms using 3×3 local neighbors. The number of propagation steps was set to 18 empirically. Other training details will be described for each dataset individually. For the quantitative evaluation, we utilized the following commonly used metrics [29, 21, 9]:

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

6.1 NYU Depth V2

The NYU Depth V2 dataset [29] (NYUv2) consists of RGB and depth images of 464 indoor scenes captured by a Kinect sensor. For the training data, we utilized a subset of ∼50K images from the official training split. Each image was downsized to 320×240, and then 304×228 center-cropping was applied. We trained the model for 25 epochs with ℓ_1 loss, and the learning rate decayed by 0.2 every 5 epochs after the first 10 epochs. We set the batch size to 24. The official test split of 654 images was used for evaluation and comparisons.

In Fig. 6, we present some depth completion results obtained for the NYUv2 dataset. As in previous works [21, 9], 500 depth pixels were randomly sampled from a dense depth image and used as the input along with the corresponding RGB image. For comparison, we provide results from the Sparse-to-Dense

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.

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.

(S2D) [21] and the CSPN [9]. The S2D (Fig. 6(c)) generates blurry depth images, as it is a direct regression algorithm. Compared to the S2D, the CSPN and our method generate depth maps with substantially improved accuracy thanks to the iterative spatial propagation procedure. However, the CSPN suffers from mixed-depth problems, especially on tiny or thin structures. In contrast, our method well preserves tiny structures and depth boundaries using non-local propagation.

Table 1 shows the quantitative evaluation of the NYUv2 dataset. The proposed algorithm achieves the best result and outperforms other methods by a large margin (RMSE 0.020m). Compared to geometry-agnostic methods [21, 19, 9], geometry-aware ones [14, 8, 25, 32] show better performance in general. The proposed algorithm can be also viewed as a geometry-aware algorithm because it implicitly explores geometrically relevant neighbors for propagation.

6.2   KITTI Depth Completion

The KITTI Depth Completion (KITTI DC) dataset [30] consists of over 90K RGB and LiDAR pairs. We ignored regions without LiDAR projection (*i.e.*, top 100 pixels) and center-cropped 1216×240 patches for training. The proposed network was trained for 25 epochs with both ℓ_1 and ℓ_2 losses to balance RMSE and MAE, and the initial learning rate decayed by 0.4 every 5 epochs after the first 10 epochs. We used a batch size of 25 for the training.

Table 2 shows the quantitative evaluation of the KITTI DC dataset. Similar to the results obtained for the NYUv2, geometry-aware algorithms [32, 25, 7, 8] perform better in general compared to geometry-agnostic methods [21, 9]. Since LiDAR sensor noise (*i.e.*, mixed foreground and background points as shown in Fig. 5) is inevitable, the predicted confidence is highly beneficial to eliminate the impact of the noise. DepthNormal [32] utilizes confidence values as a mask for weighted summation during refinement. However, its confidence mask does not totally prevent incorrect values from propagating into neighboring pixels. On the contrary, the proposed confidence-incorporated affinity normalization effectively restricts the propagation of erroneous values during propagation. We note that
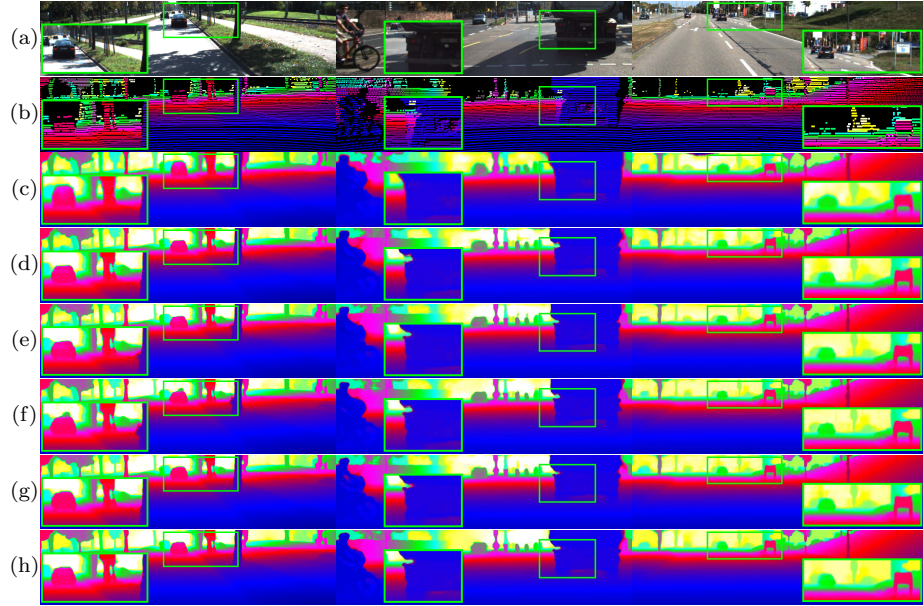
Fig. 7. Depth completion results on the KITTI DC dataset [30]. (a) RGB, (b) Sparse depth, (c) CSPN [9], (d) DepthNormal [32], (e) DeepLiDAR [25], (f) FuseNet [7], (g) CSPN++ [8], (h) Ours. Note that sparse depth images are dilated for visualization.

the proposed method outperformed all the peer-reviewed methods in the KITTI online leaderboard when we submitted the paper.

Figure 7 shows some examples of predicted dense depth with highlighted challenging areas. Those areas usually contain small structures near depth boundaries, which can be easily affected by the mixed-depth problem. Compared to the other methods (Figs. 7(c)-(g)), our algorithm (Fig. 7(h)) handles those challenging areas better with the help of non-local neighbors.

6.3   Ablation Studies

We conducted ablation studies to verify the role of each component of our network, including non-local propagation, affinity normalization, and the confidence-incorporated propagation. For all the experiments, we used a set of 10K images sampled from the KITTI DC training dataset for training and evaluated the performance on the full validation dataset. The network was trained for 20 epochs with center-cropped patches of 912×228 for fast training, and the batch size was set to 12. Other settings were set the same as those mentioned in Sec. 6.2.

Non-Local Neighbors Figure 8 visualizes some examples of non-local neighbors predicted by our algorithm. Compared to fixed-local neighbors, our predicted non-local neighbors have higher flexibility in the selection of neighbor pixels. In particular, non-local neighbors are selected from chromatically and geometrically relevant locations near the depth boundaries (*e.g.*, same objects or planes). Moreover, we collected the statistics of the depth variance of neigh-
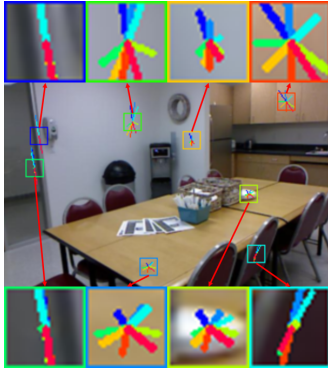
Fig. 8. Examples of non-local neighbors predicted by our network.

	Neighbors	Affinity	Norm.	Conf.	RMSE (mm)
(a)	$\mathcal{N}^{\mathrm{CS}}$	Learned	Abs$-$Sum	No	908.4
(b)				Yes	891.6
(c)			Tanh$-\gamma-$Abs$-$Sum*	No	896.4
(d)				Yes	890.4
(e)	$\mathcal{N}^{\mathrm{NL}}$	Color			930.3
(f)		Learned	Abs$-$Sum	No	903.1
(g)				Yes	889.5
(h)			Abs$-$Sum*	Yes	886.0
(i)			Tanh$-$C		886.4
(j)			Tanh$-\gamma-$Abs$-$Sum*	No	891.3
(k)				Binary	892.9
(l)				Weighted	884.8
(m)				Yes	**884.1**

Table 3. Quantitative evaluation on the KITTI DC validation set [30] with various configurations. Please refer to the text for details.

boring pixels to show the relevance of the selected neighbors. On the KITTI DC validation set, the average depth variances for fixed-local and non-local neighbor configurations were 22.7mm and 11.6mm, respectively. The small variance of the non-local neighbor configuration demonstrates that the proposed method is able to select more relevant neighbors for propagation.

The quantitative results obtained for the network with fixed-local $\mathcal{N}^{\mathrm{CS}}$ and that with non-local neighbors $\mathcal{N}^{\mathrm{NL}}$ are shown in Tab. 3. These networks were also tested with two normalization techniques: (1) with Abs$-$Sum (Tab. 3(b) and (g)), and (2) with Tanh$-\gamma-$Abs$-$Sum* (Tab. 3(d) and (m)). The proposed method with non-local neighbors consistently outperformed that with fixed-local neighbors, demonstrating the superiority of the non-local framework.

Affinity Normalization and Confidence Incorporation   To validate the proposed affinity normalization algorithm, we compare it with three different affinity normalization methods (*cf.*, Sec. 4). Table 3(g)-(i), and (m) assessed the performance using the same network but different affinity normalization methods. The model with Abs$-$Sum does not perform well due to the limited range of affinity combinations, as shown in Fig. 4(a). When relaxing the normalization condition while maintaining the stability condition (Abs$-$Sum*), the performance was improved thanks to the wider area of feasible affinity space and better affinity distribution (Fig. 4(b)). Tanh$-$C strengthens the stability condition without explicit normalization. However, as shown in Fig. 4(c), the resulting affinity values reside in a smaller affinity space (*i.e.*, in a K-dimensional hypercube with edge size $2/K$); therefore, it achieved a slightly worse performance compared to Abs$-$Sum*. The proposed Tanh$-\gamma-$Abs$-$Sum* was able to alleviate this limitation with a learnable normalization parameter γ. The learned γ compromises between Abs$-$Sum* and Tanh$-$C, and can boost the performance. Note that the final γ values (initialized with $\gamma = K = 8$) trained on the NYUv2 (Sec. 6.1) and the KITTI DC (Sec. 6.2) datasets were 5.2 and 6.3, respectively. This observation indicates that the optimal γ varies based on the training environment.

Method	CSPN [9]	DDP [33]	NConv [12]	S2D [21]	DepthNormal [32]	DeepLiDAR [25]	Ours
# Params. (M)	17.41	28.99	0.36	42.82	28.99	53.44	25.84

Table 4. Comparison of the number of network parameters. Note that only methods with publicly available implementations [9, 33, 12, 21, 32, 25] are included.

We also compared the performance of the network with and without confidence, to verify the importance of confidence incorporation. In addition, we tested two alternative confidence-aware networks (1) by generating a binary mask from confidence with a threshold of 0.5 and (2) with the weighted summation approach of DepthNormal [32], and applying each method during the propagation to eliminate the effect of outliers. The comparison results are shown in Tab. 3(j)-(m). The proposed confidence-incorporated affinity normalization (Tab. 3(m)) outperforms the others due to its capability of suppressing propagation from unreliable pixels. The mask-based (Tab. 3(k)) and weighted summation (Tab. 3(l)) approaches show worse performance compared to that of ours, indicating that the hard-thresholding and weighted summation approaches are not optimal for encouraging propagation from relevant pixels but suppressing that from irrelevant pixels. Note that the proposed confidence-incorporated approach is effective for both the network with $\mathcal{N}^{\text{NL}}$ and that with $\mathcal{N}^{\text{CS}}$ (Tab. 3(a)-(d)). These results demonstrate the effectiveness of our confidence incorporation.

Further Analysis To verify the importance of learned affinities, we further evaluated the proposed method with conventional affinities calculated based on the Euclidean distance between color intensities. As shown in Tab. 3(e) and (m), the network using learned affinities performed much better than the network using the hand-crafted one. In addition, we provide the number of network parameters of the compared methods in Tab. 4. The proposed method achieved superior performance with a relatively small number of network parameters. Please refer to the supplementary material for additional experimental results, visualizations, and ablation studies.

7   Conclusion

We have proposed an end-to-end trainable non-local spatial propagation network for depth completion. The proposed method gives high flexibility in selecting neighbors for propagation, which is beneficial for accurate propagation, and it eases the affinity learning problem. Unlike previous algorithms (*i.e.*, fixed-local propagation), the proposed non-local spatial propagation efficiently excludes irrelevant neighbors and enforces the propagation to focus on a synergy between relevant ones. In addition, the proposed confidence-incorporated learnable affinity normalization encourages more affinity combinations and minimizes harmful effects from incorrect depth values during propagation. Our experimental results demonstrated the superiority of the proposed method.

References

1. NVIDIA Apex. www.github.com/nvidia/apex
2. TESLA Autopilot. www.tesla.com/autopilot
3. UBER ATG. www.uber.com/us/en/atg
4. Barron, J.T., Poole, B.: The fast bilateral solver. In: Proc. of European Conf. on Computer Vision (ECCV) (2016)
5. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2005)
6. Chen, L.C., Barron, J.T., Papandreou, G., Murphy, K., Yuille, A.L.: Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Chen, Y., Yang, B., Liang, M., Urtasun, R.: Learning joint 2d-3d representations for depth completion. In: Proc. of IEEE Int'l Conf. on Computer Vision (ICCV) (2019)
8. Cheng, X., Wang, P., Guan, C., Yang, R.: Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In: Proc. of AAAI Conf. on Artificial Intelligence (AAAI) (2020)
9. Cheng, X., Wang, P., Yang, R.: Depth estimation via affinity learned with convolutional spatial propagation network. In: Proc. of European Conf. on Computer Vision (ECCV) (2018)
10. Chodosh, N., Wang, C., Lucey, S.: Deep convolutional compressed sensing for lidar depth completion. In: Proc. of Asian Conf. on Computer Vision (ACCV) (2018)
11. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proc. of Advances in Neural Information Processing Systems (2014)
12. Eldesokey, A., Felsberg, M., Khan, F.S.: Confidence propagation through cnns for guided sparse depth regression. IEEE Trans. on Pattern Anal. and Mach. Intell. (TPAMI) (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2016)
14. Imran, S., Long, Y., Liu, X., Morris, D.: Depth coefficients for depth completion. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Proc. of Advances in Neural Information Processing Systems (2015)
16. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: Proc. of Advances in Neural Information Processing Systems (2011)
17. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2006)
18. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2015)
19. Liu, S., De Mello, S., Gu, J., Zhong, G., Yang, M.H., Kautz, J.: Learning affinity via spatial propagation networks. In: Proc. of Advances in Neural Information Processing Systems (2017)

20. Liu, S., Pan, J., Yang, M.H.: Learning recursive filters for low-level vision via a hybrid neural network. In: Proc. of European Conf. on Computer Vision (ECCV) (2016)
21. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: Proc. of IEEE Int'l Conf. on Robotics and Automation (ICRA) (2018)
22. Park, J., Tai, Y.W., Cho, D., Kweon, I.S.: A unified approach of multi-scale deep and hand-crafted features for defocus estimation. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
24. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. on Pattern Anal. and Mach. Intell. (TPAMI) (1990)
25. Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., Pollefeys, M.: Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proc. of Int'l Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI) (2015)
27. Saxena, A., Chung, S.H., Ng, A.Y.: Learning depth from single monocular images. In: Proc. of Advances in Neural Information Processing Systems (2006)
28. Shim, G., Park, J., Kweon, I.S.: Robust reference-based super-resolution with similarity-aware deformable convolution. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2020)
29. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Proc. of European Conf. on Computer Vision (ECCV) (2012)
30. Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A.: Sparsity invariant CNNs. In: Int'l Conf. on 3D Vision (3DV) (2017)
31. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018)
32. Xu, Y., Zhu, X., Shi, J., Zhang, G., Bao, H., Li, H.: Depth completion from sparse lidar data with depth-normal constraints. In: Proc. of IEEE Int'l Conf. on Computer Vision (ICCV) (2019)
33. Yang, Y., Wong, A., Soatto, S.: Dense depth posterior (ddp) from single image and sparse range. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)
34. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. on Pattern Anal. and Mach. Intell. (TPAMI) (2006)
35. Zbontar, J., LeCun, Y., et al.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research (2016)
36. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2019)