Mask-Assisted Depth Completion with Multi-Resolution Predictions Based on Attention Mechanism from Color Image and Sparse LiDAR

Ching-Yen Shih Jing-An Tzeng Jeremy Lu Yu-Chi Wang
University of Mighigan
{cyshih, robintzg, jerlomy, yuchiw}@umich.edu

Abstract

Depth completion is essential for autonomous driving applications. In this paper, we propose the end-to-end learning architecture to effectively complete depth from a color image and sparse LiDAR data. We develop a local pathway and a global pathway to extract high-resolution features and low-resolution features respectively. The local pathway is FuseNet without 3D representation from [2], and the global pathway is made up of our proposed stacked U-Block. Our architecture combines predictions from these pathways based on the attention mechanism. With the learned confidence map, our model can put attention on local or global pathway depending on their confidence, and experiment result shows that local pathway has higher confidence on the edge, and global pathway has higher confidence inside the object. Furthermore, we apply a binary mask to help our model know positions of valid values in sparse LiDAR data, and it can boost the performance of local pathway. We evaluate our model on the KITTI depth completion dataset. To make comparison, we implement two models ranking 8^{th} and 11^{th} on the KITTI depth completion benchmark. Also, we conduct an ablation study and qualitative analysis to demonstrate the effectiveness of proposed U-Block and our methods.

1. Introduction

Depth completion aims to convert sparse depth data into a dense depth map. An accurate dense depth map is vital for a lot of robotics applications, such as autonomous vehicles and drones. Despite the fact that we can also get the high-resolution data from high-end LiDAR, those LiDARs are usually extremely expensive, which is not cost-efficient for business use. Thus, reconstructing dense depth maps from sparse LiDAR data is of great value for practical applications. There are three different kinds of common reconstruction methods, the first one is to convert a color image into a dense depth map, the second one is to convert sparse LiDAR data into a dense depth map, and the third one is to convert color image and sparse LiDAR data into a dense depth map. We choose to conduct the third method, because both of the color image and sparse LiDAR data are easily collected, and this method could provide more information for reconstructing the dense depth.

The KITTI Vision Benchmark Suite [19] is a well-known source to get the color images and sparse LiDAR data about the outdoor environment, which were collected from a moving vehicle. It also provides a platform for people to upload their methods and compare the performance of depth completion with each other. DeepLiDAR [15] and UberATG-FuseNet [2], which are ranked 11^{th} and 8^{th} on KITTI respectively, showed us some interesting ideas that turned out to be the main concept of the architecture of our neural network.

Qiu et al. [15] proposed a deep learning architecture, DeepLiDAR, which produced dense depth for the outdoor scene from a single color image and a sparse depth. This network estimated surface normals as the intermediate representation from color image sparse LiDAR data, and then estimated dense depths and confidence maps from surface normals and the color image respectively. The confidence maps are used to learn the weight on each pixel of two estimated dense depths. In the end, predicted dense depths from two pathways would be integrated into one dense depth with the help of learned confidence maps. However, DeepLiDAR has to use a lot of computational resources, and there are the large number of the parameters in the network. To be more specific, the number of parameters in DeepLiDAR is 143,981,012, which is too large to be put on the single GPU. Also, DeepLiDAR has to estimate the surface normals first, so the performance of DeepLiDAR will depend on the accuracy of surface normals estimation.

To deal with the shortage of DeepLiDAR, we design the model that can effectively predict dense depth without surface normals estimation. Furthermore, the number of parameters in our model is quite small but still can complete depth effectively. Like DeepLiDAR, our final prediction is integrated from two pathways based on attention mech-

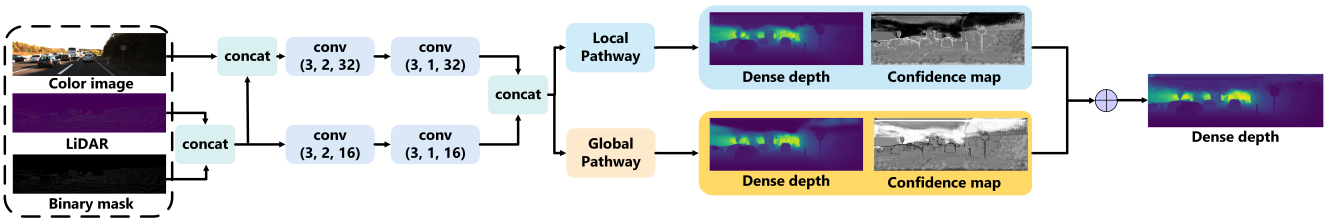


Figure 1. The pipeline of our learning architecture. Our architecture consists of a local pathway and a global pathway. Each pathway predicts the dense depth and the confidence map. We integrate predicted depth from two pathways by applying attention mechanism according to their confidence.

anism (confidence maps). However, we design the local pathway and global pathway to extract high-resolution features and low-resolution features respectively. Therefore, our model would take both kinds of features into account. Due to the attention mechanism, the weight from each pathway during integration step depends on the position in the image.

Our local pathway has the same architecture as FuseNet without 3D representation model [2]. To extract global features, we proposed U-Block. U-Block has the similar structure as U-Net, but the there are slight differences between U-Block and U-Net to ensure the feature spaces in different levels are equal. Besides, the input shape and output shape of U-Block are equal, so we can stack U-Block to build the network.

Besides, since we have to complete dense depth from sparse depth, we apply the binary mask to indicate position of valid values in the sparse LiDAR data as [15]. Applying mask has large performance improvement on our local pathway (FuseNet).

Our full pipeline is shown in Fig. 1. Our contributions are as follows. (1) We proposed U-Block to effectively extract low-resolution features, and we can build the network by stacking U-Block. (2) We apply a mask to indicate positions of valid values in the sparse LiDAR data. (3) We apply attention mechanism to let our model put attention on the local pathway (extracts high-resolution features) or global pathway (extracts low-resolution features) according to the position in the image.

2. Related Work

The depth completion algorithms can be classified as unguided depth completion or guided depth completion based on the need for RGB images.

Non-Guided Depth Completion: "Non-Guided" means RGB images are not required to perform the depth completion. Hornacek et al. [6] completed the depth map by identifying and merging patch correspondences within the input depth map itself. Uhrig et al. [20] provided a sparse convolution layer which made the network be invariant to the level of sparsity in the data. Ku et al. [11] exploited sparse LiDAR depth data and handcrafted classical image processing algorithms to product the dense LiDAR data.

Guided Depth Completion: "Guided" means RGB images are required to perform the depth completion. Chen et al. [1], Qi et al. [14], and Richardt et al. [16] completed the depth map with bilateral filtering while Matyunin et al. [13] used median filters. Recently, deep learning shows a great success on guided depth completion like Hui et al. [9], Song et al. [17]. Eigen et al. [4] proposed a multi-scale deep convolutional network to predict depth from coarse to fine from a single RGB image. Tang et al. [18] design the guided convolution neural network and generates content-dependent and spatially-variant kernels to guide the sparse LiDAR data through RGB image. Generally, guided depth completion achieves better results because including an additional RGB image means more clues to deduce the missing depth value. Moreover, the first ranked method on KITTI currently is GuideNet [18], belonging to guided depth completion. This fact gave us the motivation to take a color image as an input.

Convolutional Neural Networks (CNNs) like Chodosh et al. [3], Hua et al. [7], Uhrig et al. [21] show excellent performance on depth completion. However, because the common datasets for scene depth completion such as the KITTI-Depth dataset [19] have a very high spatial resolution, the state-of-the-art methods Huang et al. [8], Jaritz et al. [10], Ma et al. [12] demand huge CNNs with millions of parameters. In other words, the implementation requires lots of computational and memory resources, which boosts the difficulty of implementing the algorithms in autonomous driving and robotic systems. Chen et al. [2], Eldesokey et al. [5] handle such a problem by designing a new layer or block with a lower number of parameters. In our network, "U-Block" we designed to effectively reduce the demand for memory.

3. Method

We develop the learning architecture (as illustrated in Fig. 1) that can effectively complete the depth from a color image and sparse LiDAR data. Our model consists of two pathways: the local pathway and the global pathway. The local pathway has the same structure as FuseNet without 3D

representation module. The local pathway aims to extract high-resolution features, and it is made up of 2D block [2], which is illustrated in Fig. 2(b). The global pathway extracts low-resolution features, and it comprises our proposed U-Block, as shown in Fig. 2(a). Also, we improve the performance of the local pathway by concatenating binary mask to the sparse LiDAR data, because the binary mask can help our model to indicate the valid values of sparse LiDAR data. Finally, to combine the results of the local and global pathways, we apply the attention mechanism to integrate the predicted dense from two pathways.

3.1. U-Block

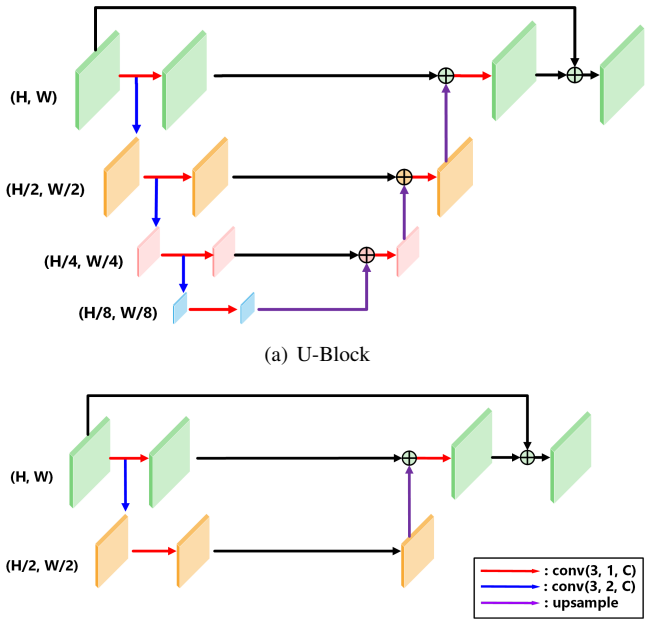
Our global pathway is built based on the U-Block. We develop U-Block to extract low-resolution features as shown in Fig. 2(a). Because of four downsampling layer in the U-Block, it can extract features with lower resolution and higher receptive field than 2D block which is used in the local pathway. To ensure feature map with different resolution in the same space, unlike U-Net, we use bilinear upsampling rather than upconvolution as our upsampling layer, and use addition rather than concatenation to integrate feature maps with the same level. Since bilinear upsampling uses interpolation to upsample the feature map, so the feature space will not be changed. And when we add two quantity, there will be meaningful if the two quantity are in the same space, so using addition in UBlock can ensure our feature maps in different level are in the same space. Also, the input and output of the block is a feature map with shape $C \times H \times W$, so we can directly stack our U-Block to build the network.

3.2. Local Pathway and Global Pathway

We complete depth by integrating predicted depths from local pathway and global pathway. The structure of the pathway is shown in Fig. 3. The local pathway consists of the stacked 2D block and it extracts the high-resolution features to generate dense depth. Therefore, this pathway focuses on the local features. On the other hand, the global pathway contains stacked U-Block. It can extract low-resolution features to complete depth. These two pathway has the complement effect, the one focuses on a small-scale features, and the other one focuses on large-scale features. Also, the two pathways generate the confidence mask to indicate their prediction confidence on each pixel.

3.3. Binary Mask

We use a pixel-wise binary mask to indicate the availability of sparse LiDAR data. By providing a binary mask, our model can easily point out where the valid values of LiDAR data are, and then more rely on these values. The expression of mask is shown as follows



(a) U-Block

(b) 2D Block

Figure 2. (a) Structure of U-Block. This module is to extract low-resolution features. We use bilinear upsampling to upsample the smaller feature map and use addition to integrate feature maps with the same level. (b) Structure of 2D block. This module aims to extract high-resolution features. $conv(K, S, C)$ denotes 2D convolution with kernel size K, stride S and output channels C.

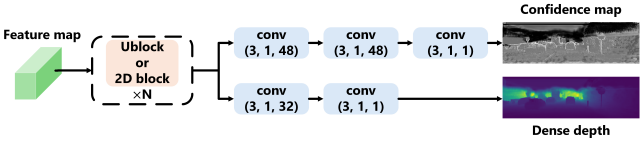


Figure 3. Structure of the pathway. Local pathway is stacked by 2D block, and global pathway is stacked by U-Block.

$$M(x,y) = \begin{cases} 1, & \text{if } L(x,y) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $M(x,y)$ is the mask value on position (x,y) and $L(x,y)$ is the value of sparse LiDAR data on position (x,y)

3.4. Attention Mechanism

To integrate the results from the local pathway and global pathway, we apply the attention mechanism to sum the weighted depths from two pathways. We generate the confidence maps from both the local pathway and global pathway and then use pixel-wise softmax to get the weights of the pixel in each pathway. Therefore, the model would choose to focus on the local pathway or global pathway according to their confidence. The final depth prediction is as

follows

$$D = w_g \cdot D_g + w_l \cdot D_l \tag{2}$$

where D is the predicted dense, w is the weight of the pixel, and subscript l and g represents local pathway and global pathway.

4. Experiments

We conduct our experiment on KITTI depth completion dataset. To make the comparison, we implement two models ranking 8^{th} and 11^{th} on the KITTI depth completion benchmark. In addition, we do ablation studies and qualitative analysis to demonstrate the effectiveness of our methods.

4.1. Experiment Setting

4.1.1 Dataset

The KITTI dataset has been recorded from a moving wagon equipped with four video cameras (two color and two grayscale cameras), a rotating 3D laser scanner and a combined GPS/IMU inertial navigation system. The main objective of this dataset is to give an impetus to the development of the self-driving car, especially in computer vision and robotic algorithms. The KITTI depth completion dataset contains 85,898 frames for training, 1,000 frames for validation, and another 1,000 frames for testing. Also, accurate ground truth is provided by a Velodyne laser scanner and a GPS localization system. In consideration of training time, we only use 20,000 training data and we randomly cropped the input image of both testing and training into the size of 128×256.

4.1.2 Metrics

In depth completion, there are four standard metrics for evaluation, namely, root mean squared error (RMSE), mean absolute error (MAE), root mean squared error of the inverse depth (iRMSE), and mean absolute error of the inverse depth (iMAE). Among them, we decide RMSE as our metrics since RMSE is more sensitive and is selected as the dominant metrics for KITTI leaderboard in the depth completion benchmark.

4.1.3 Implementation details

Loss Function The loss function of the system is defined as:

$$L = \lambda_1 L_d(D_l) + \lambda_2 L_d(D_g) + \lambda_3 L_d(D_m) \tag{3}$$

where D_l, D_g, D_m, indicate the predicted depth of local pathway, global pathway, and merged result of two pathways. For L_d, which defines the loss on the estimated depth, we adopt mean squared error (MSE) or so-called l_2 loss to compute the loss function between ground truth and predicted depth. λ manipulates the weight between terms of loss function. In our experiment, we set $\lambda_1 = 0.25$, $\lambda_2 = 0.25$, $\lambda_3 = 0.5$ to train the whole system.

Training Setting We utilize Adam as the optimizer with learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, the number of 2D blocks in the local pathway is 12, the output channel for the convolution kernel in the 2D block and U-Block is 64. In addition, We use a GTX 1080 GPU for training with batch size 8. We train our model from the ground up without any pretrained model and employing extra datasets.

4.2. U-Block

To validate the effectiveness of U-Block, we compare performance between 2D block (used in local pathway) and U-Block (used in global pathway) as shown in Table. 1. The experiment shows that U-Block performs better than 2D block because it can extract features with larger receptive field by more downsampling layers. Also, we compare addition (`Ublock_add`) with concatenation (`Ublock_cat`) when we merge features in the same level. As shown in Table. 2, addition outperforms concatenation a lot. It is because we use bilinear upsampling to upsample feature maps, the space of feature maps in different levels are the same. Therefore, addition can merge information of features with the same level more effectively.

Method	RMSE (mm)
`2D block (Local pathway)`	1152.8913
`U-Block (Global pathway)`	**1070.5581**

Table 1. Comparison of the performance between 2D block and U-Block.

Method	RMSE (mm)
`Ublock_cat`	1346.0962
`Ublock_add`	**1010.5581**

Table 2. Comparison of the performance between concatenation and addition to merge information of feature maps on the same level in U-Block.

4.3. Mask

The binary mask can provide our model the availability of sparse LiDAR data. In this subsection, we compare two different mask usage on the global pathway as shown in Table. 3. The first one, `Concat_mask`, is concatenating the mask to the LiDAR data before feeding it to the model. The second one, `Lidar+Model`, simply adds the results from sparse LiDAR data and our model with inverted mask. The equation of second method is as follows

$$D = P \times inv\_mask + L \times mask \tag{4}$$

where D is the final predicted depth, P is the predicted depth from our model, L is sparse LiDAR data, $mask$ is the binary mask to indicate the position of valid sparse LiDAR depth, and inv_mask is the pixel-wise inversion of $mask$.

It is obvious that `Concat_mask` outperforms `Lidar+Model` a lot because there is an occlusion between the LiDAR sensor and Velodyne laser (groundtruth). Therefore, directly outputting the valid values from sparse LiDAR data will be inaccurate.

Mask	RMSE (mm)
`Lidar+Model`	1282.9781
`Concat_mask`	**1035.9539**

Table 3. Comparison between two different kinds of methods to apply the binary mask

4.4. Attention Mechanism

The local pathway and the global pathway extracts the features under distinct resolution. Therefore, we utilize the attention mechanism to integrate results from both pathways. The mechanism lets our model to learn which pathway produces more reliable depth data for each area. The experiment result shows that integrating two pathways outperforms any individual pathway as shown in Table. 4. Moreover, we visualize the confidence maps from both pathways as shown in Fig. 4. The confidence map from the local pathway shows higher confidence on the edge. On the other hand, the confidence map from the global pathway shows higher confidence inside the object.

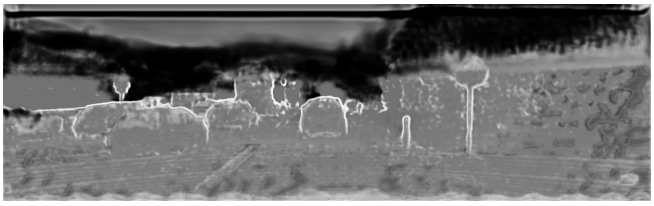
4.5. Ablation Studies

We conduct ablation studies to show the effectiveness of our methods as shown in Table. 4. First, applying the binary mask can effectively improve performance on the local pathway, but the mask is not helpful for the global pathway. Second, we use attention mechanism to integrate results from global pathway and local pathway, it can effectively boost the performance and it shows that these two pathways have the complement effect.

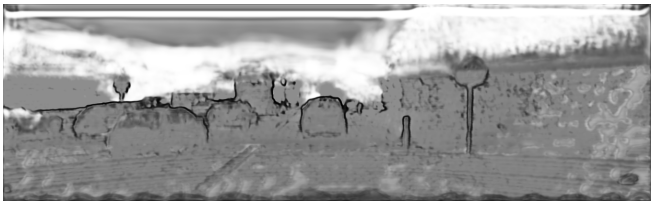
Global pathway	Local pathway	Mask	RMSE (mm)
✓			1070.5581
✓		✓	1093.1014
	✓		1152.8913
	✓	✓	1035.9539
✓	✓	✓	**1005.4614**

Table 4. Ablation studies of depth completion on KITTI validation dataset

(a) Color image



(b) Confidence map from the local pathway



(c) Confidence map from the global pathway

Figure 4. (a) Original color image. (b) Confidence map from local pathway. Local pathway has higher confidence on the edge. (c) Confidence map from global pathway. Global pathway has higher confidence inside the object.

4.6. Comparison

To make the comparison, we implement two models ranking 11^{th} and 8^{th} on the leader board of the KITTI depth completion benchmark. The first one is a light version of DeepLiDAR[15][1]. The reason why we don't use the original DeepLiDAR is that the original one is too large to put onto a single GPU we have. The number of parameters in original DeepLiDAR is 143,981,012. Although we implement the light version of DeepLiDAR, the number of parameters is still really large, i.e., 47,750,544. The second model we implemented is FuseNet[2] without 3D representation branch.

In the network we design, the number of parameters is 5,687,044, which is much smaller than DeepLiDAR but our model has the better performance. Also, our model can be trained without the estimation of surface normals. Our local pathway uses the same architecture as FuseNet while the global pathway replaces the 2D block in the FuseNet with our proposed U-Block. The experiment results show that the global pathway has the better performance than the local pathway. In other words, it shows that our proposed U-

[1]Our implementation of DeepLiDAR with python3 and the newest version of Pytorch: https://github.com/ChingYenShih/DeepLidar

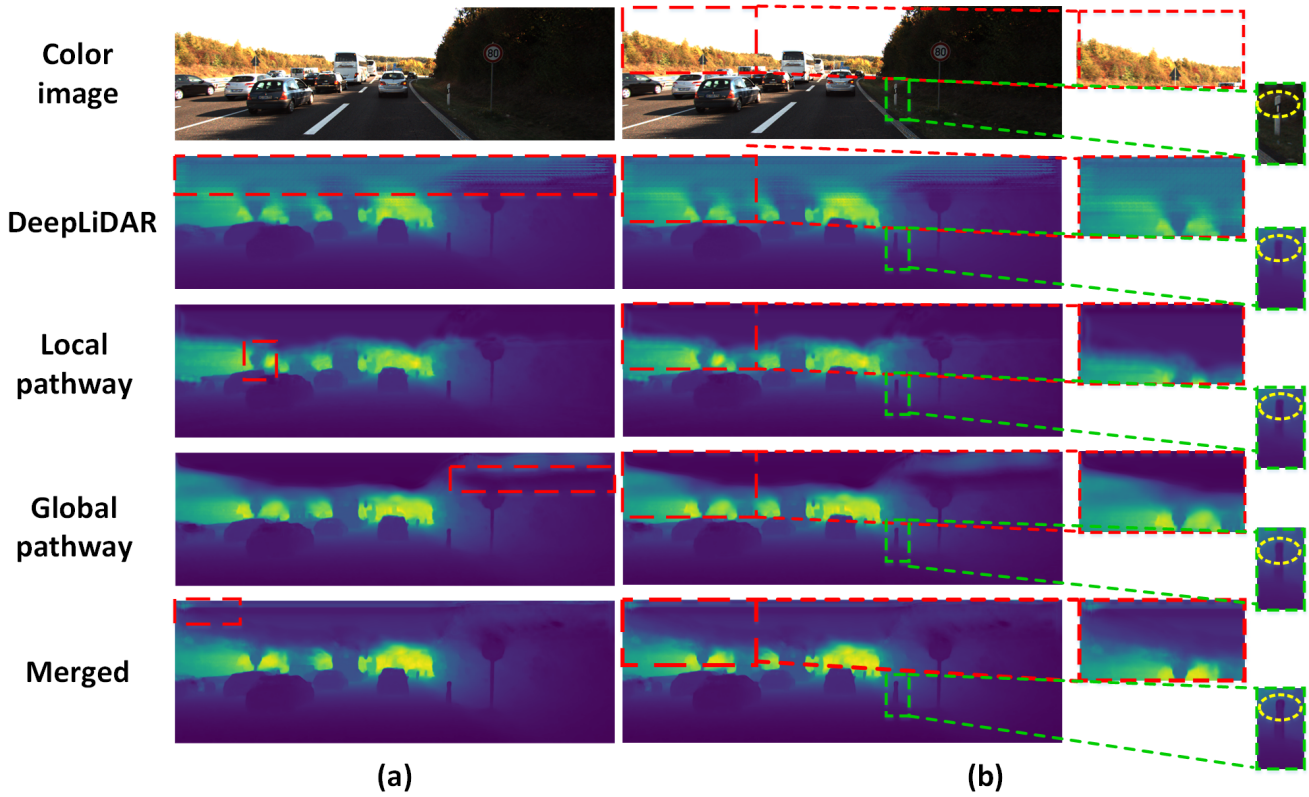


Figure 5. The qualitative analysis of different method. Column (a) indicate obviously abnormal area in each method. Column (b) compares performance of each method in the same position.

Block is better than the 2D block in the FuseNet. Besides, by applying attention mechanism, we can further improve the performance.

Method	RMSE (mm)
DeepLiDAR[15]	1191.6127
FuseNet[2]	1152.8913
Global pathway (U-Block)	1070.5581
Integrate two pathways	**1005.4614**

Table 5. Comparison between our method and others

4.7. Qualitative analysis

We perform qualitative analysis on each method as shown in Fig. 5. In column (a), we indicate obviously abnormal area in each method. The sky region performed by DeepLiDAR has lots of noise, and the local pathway doesn't perform well in the detailed region. The global pathway and merged result also have slight noise on the tree and sky respectively.

In column (b), we compare the performance of each method in the same location. The red region has high-frequency object (leaves) and there is a boundary between the tree and sky in this region. DeepLiDAR perform the worst, the red area is quite noisy. For local pathway, the boundary is obscure, but it can successfully capture high-frequency area. For global pathway, the boundary and high-frequency object are both smoother than the local pathway. For merged pathway, it combines the characteristics of local and global pathway. The green region contains a bar-like object. The DeepLiDAR and local pathway cannot capture the "notch" (yellow circle) on the bar, but our proposed global pathway and merged result can capture the "notch" (yellow circle).

5. Conclusion

We propose the learning architecture to effectively complete depth from a color image and sparse LiDAR data. First, we propose U-Block to capture low-resolution features and it perform better than 2D block [2]. Second, we improve the performance of the local pathway [2] by applying a binary mask to indicate the valid values in sparse LiDAR data. Third, we integrate results from the local and global pathway by attention mechanism, and it not only has better performance but also combines characteristics of the local and global pathway.

References

[1] Li Chen, Hui Lin, and Shutao Li. Depth image enhancement for kinect using region growing and bilateral filter. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3070–3073. IEEE, 2012.

[2] Yun Chen, Bin Yang, Ming Liang, and Raquel Urtasun. Learning joint 2d-3d representations for depth completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10023–10032, 2019.

[3] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. In *Asian Conference on Computer Vision*, pages 499–513. Springer, 2018.

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[5] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *arXiv preprint arXiv:1811.01791*, 2018.

[6] Michael Hornacek, Christoph Rhemann, Margrit Gelautz, and Carsten Rother. Depth super resolution by rigid body self-similarity in 3d. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1123–1130, 2013.

[7] Jiashen Hua and Xiaojin Gong. A normalized convolutional neural network for guided sparse depth upsampling. In *IJCAI*, pages 2283–2290, 2018.

[8] Zixuan Huang, Junming Fan, Shenggan Cheng, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *IEEE Transactions on Image Processing*, 2019.

[9] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European conference on computer vision*, pages 353–369. Springer, 2016.

[10] Maximilian Jaritz, Raoul De Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. In *2018 International Conference on 3D Vision (3DV)*, pages 52–60. IEEE, 2018.

[11] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018.

[12] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019.

[13] Sergey Matyunin, Dmitriy Vatolin, Yury Berdnikov, and Maxim Smirnov. Temporal filtering for depth maps generated by kinect depth camera. In *2011 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2011.

[14] Fei Qi, Junyu Han, Pengjin Wang, Guangming Shi, and Fu Li. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, 34(1):70–76, 2013.

[15] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] Christian Richardt, Carsten Stoll, Neil A Dodgson, Hans-Peter Seidel, and Christian Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos. In *Computer graphics forum*, volume 31, pages 247–256. Wiley Online Library, 2012.

[17] Xibin Song, Yuchao Dai, and Xueying Qin. Deep depth super-resolution: Learning depth super-resolution using deep convolutional neural network. In *Asian conference on computer vision*, pages 360–376. Springer, 2016.

[18] Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. *arXiv preprint arXiv:1908.01238*, 2019.

[19] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[20] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20. IEEE, 2017.

[21] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger. Sparsity invariant cnns. In *2017 International Conference on 3D Vision (3DV)*, pages 11–20, 2017.