[1] **Lu-Wen Chen**
[2] **Wei-Jong Yang**

A Depth Completion Network for Efficient Depth Representation

Abstract: - 3D information with depth pixels or Lidar points is the key component for robotic perception, autonomous driving, virtual reality and 3D movies. Traditional 3D data representation requires large amounts of data, making the transmission bandwidth and storage resource inefficiency. In this paper, instead of complete 3D information, we only need a few of 3D depth information and use depth completion techniques to improve data efficiency by predicting the missing depth data by leveraging sparse depth map and RGB image to reconstruct dense depth map. We can reduce the amount of depth data while maintaining quality and accuracy in the receiver by using a depth completion network. The proposed depth completion network combines autoencoder and guided attention mechanisms to enhance depth representation efficiency. The proposed approach reconstructs more accurate depth maps than the other approaches if we use the same number of depth pixels to reduce the data transmission burden while ensuring high-quality depth information. The proposed approach demonstrates the practical and efficient application of depth completion.

Keywords: Depth Completion, Autoencoder, Guided-Attention, Depth Sparsity

I. INTRODUCTION

In recent years, the development of 3D broadcasting systems has gained significant attention due to their potential to revolutionize fields like entertainment, virtual reality, and autonomous driving. Traditional 3D data transmission requires large amounts of data, which is time-consuming and resource-intensive. Depth completion techniques aim to predict and fill missing depth information from depth sensors, creating accurate depth maps. By using sparse depth maps and RGB images, these models can reconstruct dense depth maps, reducing data transmission needs while maintaining quality. Deep learning and neural networks are crucial in depth completion. Some studies use convolutional neural networks (CNNs) to extract features and reconstruct dense depth maps, while others use Transformer-based architectures for enhanced accuracy and efficiency in depth prediction.

Depth completion technology reduces the burden of 3D data transmission by using sparse depth maps and RGB images to reconstruct high-precision dense depth maps. This optimizes data transmission while maintaining quality and accuracy. The key lies in effectively sampling and utilizing sparse depth points. Current research focuses on developing algorithms and models to predict missing depth information from limited data. Deep learning and neural networks show great potential, providing robust results. Thus, exploring and optimizing depth completion technology holds significant academic and practical value.

Our research aims to advance this technology by developing sophisticated models and effective depth sampling strategies. By optimizing how sparse depth points are sampled, we aim to reconstruct more accurate depth maps while transmitting the same number of points, enhancing efficiency and reducing the data transmission burden while ensuring high-quality depth information.

II. RELATED WORKS

A. RGB-based Depth Estimation

Stereo matching methods [1], [2], requiring dual-view images can calculate disparity to obtain depth maps while monocular depth estimation, using a single RGB image is simpler but more challenging. Monocular depth estimation employs deep learning, divided into supervised and unsupervised methods. Supervised learning [3] generates a coarse depth map refined by a fine-scale network. Unsupervised learning [4] uses left-view images to compute disparity consistency without annotated ground truth, reducing costs. Depth prediction from a single RGB image is ill-posed due to infinite possible depth maps. The proposed method integrates sparse depth input into camera-only approaches to reduce ambiguity and improve accuracy.

B. Depth Completion

With sparse depth pixels, depth completion methods aim to recover dense depth maps by using supervised deep learning networks [5]-[8], where the LiDAR sensors could provide sparse depth values. Multi-branch networks [8]-[9] have been adopted for multimodal fusion, using RGB images combined with sparse depth maps to produce high-

[1] Institute of Computer and Communication Engineering, National Cheng Kung University, Tainan. Email: jason.410260@gmail.com
[2] Department of Artificial Intelligence Applications, National Chin-Yi University of Technology, Taichung. Email: wjyang@ncut.edu.tw

precision dense depth maps. SparseFormer [10] leverages the global attention [11] [12] mechanism of transformers to process extremely sparse depth data. GuideFormer [11] is a depth completion framework that utilizes a guided-attention mechanism to fuse multimodal information from RGB images and sparse depth maps. To avoid global self-attention, GuideFormer adapts the normal self-attention concept [14] into guided-attention where image features guide the attention for sparse depth features.

III. PROPOSED METHODS

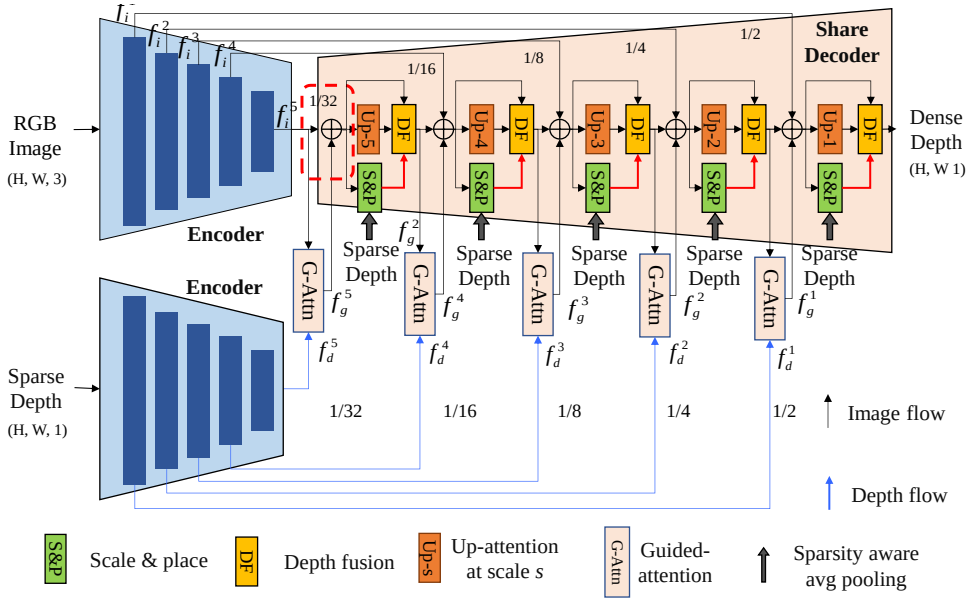

Fig. 1. Framework of proposed depth completion network

As shown in Fig. 1, the proposed dual-branch depth completion network initially involves feature extraction from RGB image and sparse depth to obtain image features f_i^s and depth features f_d^s, where superscript, *s* denotes the scale index. In Fig.1, there are five scales passed to the shared decoder via direct and skip connections for feature fusions.

The depth feature at the *s*-scale f_d^s will not directly add to the other features before being integrated into the shared decoder. Instead, it first passes through guided-attention (G-Attn) to become the guided feature f_g^s as detailed in GuideFormer [11], which is then used for feature fusion. Here, the output of depth fusion (DF), named the fused feature f_f^s, serves as another input feature to guide the depth feature f_d^s, resulting in the output guided feature f_g^s.

The fused feature f_f^s produced by depth fusion (DF) is element-wise added with the other two features, f_i^s and f_g^s to form the main feature f_m^s, which serves as the primary input for three blocks: up-attention at scale *s* (Up-*s*), scale and place (S&P), and depth fusion (DF). Detailed descriptions of these blocks are provided later.

The up-attention at scale *s* (Up-*s*) block outputs the up-sampled feature f_u^{s-1}, with f_m^s as its input. Simultaneously, f_m^s is input into the scale and place (S&P) block, which utilizes the sparse depth's available points to obtain the scaled and placed depth $\hat{d}^s$ at the corresponding scale. To propagate the information from the scaled and placed depth map $\hat{d}^s$ forward, we introduce a depth fusion (DF) block to fuse the main feature f_m^s, resulting in the fused feature f_f^s.

Finally, due to the unique nature of the smallest scale features, shown in the red dashed box in Fig. 1, we discuss it separately. At $s=5$ scale, the features are not converted into depth maps; the DF and S&P blocks for adjusting the predicted depth are unnecessary. The fused feature f_f^5 is replaced by image features f_i^5, which serves as the

guiding feature input for G-Attn. The f_m^5 is obtained by element-wise adding f_i^5 and f_g^5, and then serves as the input for Up-5, DF, and S&P blocks.

A. Encoder

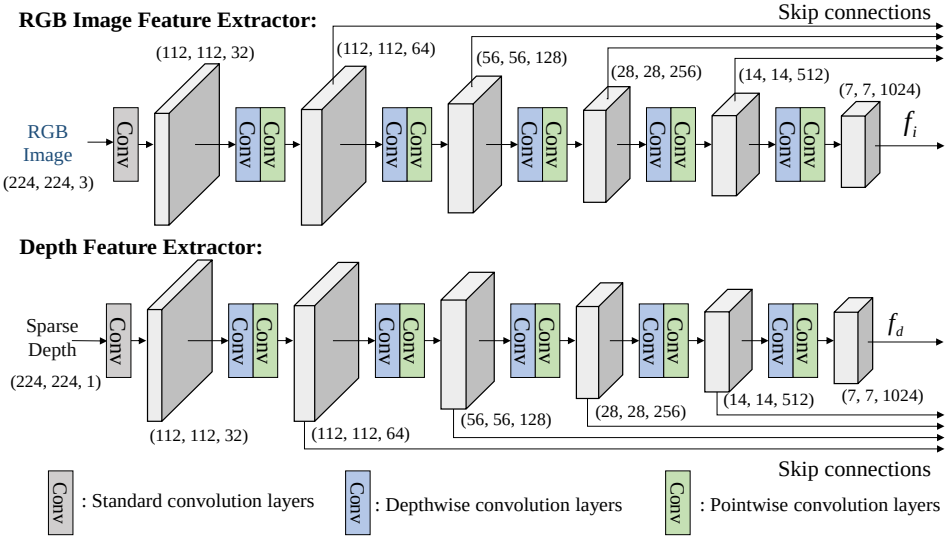

Fig. 2. Flow chart of the dual-path encoder with MobileNet-s [24]

As shown in Fig. 2, we adopt the simplified MobileNet, called MobileNet-s [15] as the encoder, where the original MobileNet [16] is revised to create a lightweight dual-path feature extractor designed for efficiency, suitable for mobile and low-latency applications. The inputs to the network are RGB images of size $224\times224\times3$ and corresponding sparse depth map of size $224\times224\times1$ with only a few depth pixels.

As shown in Fig. 2, the color image and depth extraction branches begin with a standard convolution layer followed by five sets of depthwise and pointwise convolutions. For the first set, the depthwise convolution has a stride of 1, ensuring that the spatial dimensions remain unchanged while doubling the number of channels by the pointwise convolution. For the subsequent sets, the depthwise convolutions have a stride of 2 to reduce the spatial dimensions, while the pointwise convolutions have a stride of 1 to double the number of channels.

At the end of the encoder, each branch outputs feature maps at multiple scales. The final outputs, at $1/32$ of the original size, are f_i^5 for the RGB feature extractor and f_d^5 for the depth feature extractor. Additionally, feature maps at $2^{-1}, 2^{-2}, 2^{-3}$, and 2^{-4} of the original size are fed into the shared decoder through skip connections to preserve local structure and depth information.

B. Proposed Share Decoder

The proposed shared decoder is designed to fuse the features from RGB images and sparse depths to reconstruct a dense depth map. This decoder processes the features with 5 scales producing corresponding depth map predictions and their confidence maps. The shared decoder with guided attention mechanisms ensures that both RGB contextual information and depth-specific details are effectively combined. For each scale *s*, the decoder generates estimated depth maps and their associated confidences to refine the depth estimation scale by scale. We will explain up-attention (Up-*s*), scale and place (S&P) and depth fusion (DF) blocks in following subsections.

1. Up-Attention Block

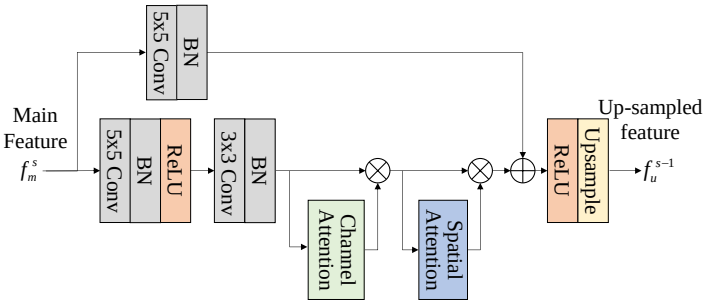

Fig. 3. Architecture of up-attention block

As shown in Fig. 3, the up-attention block at scale *s* (Up-*s*) can enhance feature map resolution while preserving depth information. For $s = 0, 1, ..., 4$, the up-attention block inputs the main feature f_m^s. The output is the upsampled feature f_u^{s-1} from the previous scale, where f_m^s passes through two parallel 5×5 convolutional layers with batch normalization (BN). One goes through ReLU, a 3×3 convolution, BN, and ReLU and then passes through channel attention and spatial attention mechanisms to enhance the important features. The attention-enhanced maps merge with the direct 5×5 convolution output through element-wise addition. The merged map [17] is upsampled by using ReLU and nearest neighbor interpolation to produce f_u^{s-1}.

2. Scale and Place (S&P) Block

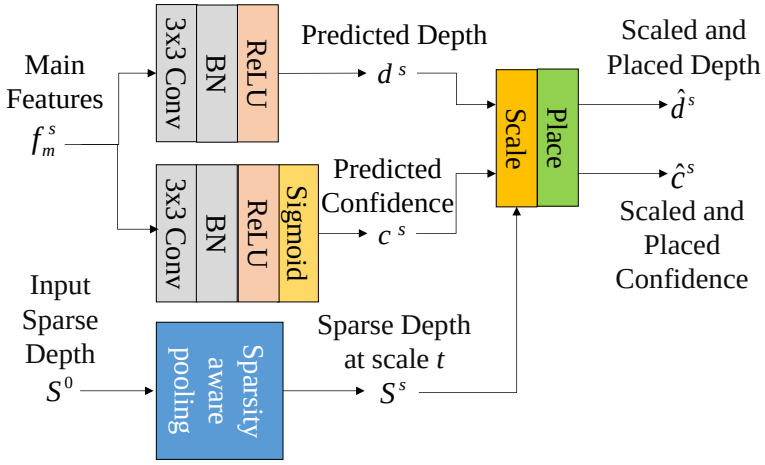

Fig. 4. Architecture of scale and place module (S&P)

As illustrated in Fig. 4, the scale and place (S&P) block is inspired by SpAgNet [18] with some modifications to fit our needs. We utilize the input sparse depth and predicted confidence to predict dense depth map and replace the original points of depth map with the available depth inputs.

In the upper branch, the main features f_m^s pass through a 3×3 convolutional layer, followed by batch normalization (BN) and a ReLU activation function to produce the predicted depth d^s. In another branch, in parallel, the main features f_m^s pass through another 3×3 convolutional layer, followed by batch normalization (BN) and a Sigmoid activation function to produce the predicted confidence c^s.

To obtain sparse depth maps at different scales, the input sparse depth S^0 is processed using the sparsity-aware pooling method. This involves moving a 3×3 window with a stride of 2 across the input, assigning the mean value of the available depth values within the window to each coordinate. This process is repeated iteratively to achieve lower resolutions, resulting in the scaled sparse depth S^s.

In the *Scale* step, the S&P module adjusts the predicted depth by performing a weighted linear regression, considering the available sparse input points and their associated confidence values. The parameters for this regression can be calculated directly and are differentiable as

$$\beta = \frac{\sum_i c_i (d_i - d')(s_i - s')}{\sum_i c_i (d_i - d')}, \quad \alpha = s' - \beta d' \tag{1}$$

With

$$d' = (\sum_i c_i d_i)/(\sum_i c_i), \quad s' = (\sum_i c_i s_i)/(\sum_i c_i) \tag{2}$$

where d_i is the predicted i^{th} depth value, c_i and s_i denote the confidence and corresponding input i^{th} sparse depth value, respectively. The scaled depth $\tilde{d}$ is then computed as $\tilde{d} = \alpha d$.

In the place step, we replace the values in the scaled depth map $\tilde{D}^s$ with the available sparse input depth values S^s at scale *s*. Additionally, the confidence map C^s is updated to reflect the highest confidence for these points as

$$\hat{D}^s[x,y]=\begin{cases}\tilde{D}^t[x,y] & \text{if } S[x,y]=0\\ S^s[x,y] & \text{if } S[x,y]\neq 0\end{cases} \tag{3}$$

$$\hat{C}^s[x,y]=\begin{cases}C^s[x,y] & \text{if } S[x,y]=0\\ 1 & \text{if } S[x,y]\neq 0\end{cases} \tag{4}$$

The S&P block ensures that the predicted dense depth maps are precisely adjusted based on the predicted confidence and the original sparse depths.

Depth Fusion Block

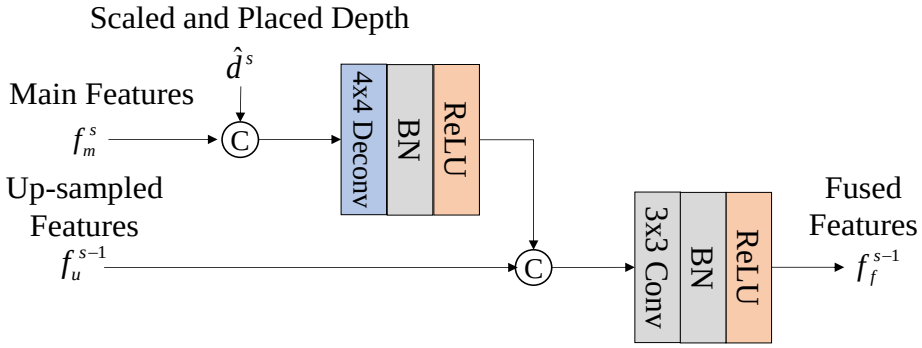

Fig. 5. Architecture of depth fusion block (DF)

Inspired by BPNet [19], the depth fusion (DF) block as shown in Fig. 5 integrates depth information into the feature maps for subsequent layers. Unlike SpAgNet, which utilizes the scaled and placed depth maps solely for calculating different scale losses without propagating them through the network, the proposed depth fusion (DF) block ensures the continuous integration of depth information throughout the network.

In the DF block, the main features f_m^s are combined with the scaled and placed depth $\hat{d}^s$ through element-wise addition. The resulting combined features are then upsampled using a 4×4 deconvolution layer, followed by batch normalization (BN) and a ReLU activation function. The upsampled features f_u^{s-1}, which are the output of the s-scale up-attention block, are then concatenated with the previous integrated features, as shown in Fig. 5. The concatenated features are further processed through a 3×3 convolutional layer and BN and a ReLU activation function, resulting in the fused features f_f^{s-1}. The continuous integration of depth information at each stage ensures that the decoder network retains and utilizes known spare depth data effectively to produce accurate depth maps.

Training Loss Function

We employ a multi-scale loss function to optimize the network. In this context, C^s and D^s represent the confidence and depth at scale *s*, respectively. The loss function incorporates the L2 loss, which measures the squared difference between the predicted depth D^s and the ground truth depth D^{GT}:

$$L_i^2=\left|\mathbf{U}_t(D^s)_i-D_i^{GT}\right|^2 \tag{5}$$

The bilinear interpolation operation is used to ensure that the predicted depth maps at different scales match up with the resolution of the ground truth depth map. The overall loss function is defined as:

$$L=\sum_{s=0}^{4}\gamma^s\frac{1}{N}\sum_{i\in V}C_i^sL_i^2-\eta\ln C_i^s \tag{6}$$

where N denotes the number of valid pixels, and $i\in V$ denotes the set of valid pixels. The hyper-parameter γ^s assigns weights to different scales, with lower scales receiving less weight. The regularization term η encourages the network to maintain high confidence values, promoting more reliable predictions. Here, we set γ=0.4 and η=0.1. By incorporating multi-scale loss and confidence weighting, the training process ensures that the network learns to produce accurate depth maps at various resolutions while maintaining high confidence in predictions. It is important to note that the loss is computed before the Place step, ensuring that the network is trained to predict both accurate depth values and reliable confidence estimates.

IV. EXPERIMENTAL RESULTS

Datasets

The datasets used for training and testing are from the NYU Depth V2 dataset [20], a widely used indoor dataset for depth completion. This dataset contains 464 indoor scenes captured using a Microsoft Kinect sensor, providing approximately 50,000 RGBD images with an original resolution of 640×480. We downsampled each image and depth map to 320×240, then center-cropped them to 224×224 pixels to prepare the data for training and evaluation. For training, we used approximately 50,000 images sampled from 249 scenes. We used the official test set, which includes 654 images from 215 scenes. During training, we sampled different numbers of points from the ground truth depth map, depending on the specific requirements of experiments, to generate sparse depth inputs.

Training Setting

We conducted our experiments using an NVIDIA GeForce RTX 3080 Ti 12GB GPU and an Intel Core i7-11700K CPU. The software environment included PyTorch 2.0.1 and Python 3.10. For training, we used AdamW [21] optimizer with an initial learning rate set to 0.0002 and a weight decay of 0.01. The batch size was 10, and the model was trained for 100 epochs with a learning rate decay applied every 10 epochs with a decay factor of 0.6.

Ablation Studies

To evaluate the effectiveness of different blocks in the decoder, we conducted ablation studies on three key components: Scale and Place (S&P), Depth Fusion (DF), and Guided Attention (G-Attn). For each configuration, the model was retrained from scratch to ensure fair comparisons. When S&P is removed, we still retain its depth and confidence prediction heads to obtain the depth map and its corresponding confidence at each scale for loss calculation. The predicted depth is also used as the depth input for the DF block. The results as shown in Table 1 illustrate the impact of these components on the model's performance.

TABLE I. Ablation studies on the effectiveness of different blocks

S&P	**DF**	**G-Attn**	**RMSE ↓**	**REL↓**	δ_1	δ_2	δ_3
✓			0.305	0.069	94.66	98.90	99.69
✓	✓		0.286	0.061	95.15	<u>98.99</u>	**99.75**
	✓		0.312	0.069	<u>94.57</u>	98.57	99.32
		✓	0.280	0.065	95.12	98.63	99.27
	✓	✓	0.292	0.061	95.40	98.87	99.49
✓		✓	<u>0.276</u>	<u>0.060</u>	95.41	98.90	99.63
✓	✓	✓	**0.270**	**0.058**	**95.83**	**99.11**	<u>99.74</u>

Table 1 shows that the combination of all three (S&P, DF, and G-Attn) blocks provides the best performance across all evaluation metrics. Specifically, when examining the scenarios where the S&P block is used, the presence of the DF block significantly impacts overall performance. This indicates that integration of depth maps into the feature fusion process, especially after scaling and placing (S&P), greatly enhances the accuracy of the depth map reconstruction. The fusion process is vital for achieving precise depth completion.

TABLE II. Ablation studies on the up-attention block at scale s (Up-s)

Up-*s*	**RMSE↓**	**REL↓**	δ_1	δ_2	δ_3
w/o attention	0.276	0.057	**95.92**	99.09	99.73
w/ attention	**0.270**	**0.058**	95.83	**99.11**	**99.74**

To evaluate the performance of our model with and without the channel and spatial attention mechanisms in the up-attention block at scale *s* (Up-*s*), we conducted ablation studies. The results presented in Table 2 show the impact of these attention mechanisms on the model's performance.

The channel and spatial attention mechanisms in the up-attention block improves the model's performance, as evidenced by lower RMSE and REL values and higher δ_1, δ_2, and δ_3 values. The attention mechanisms enhance the model's depth completion accuracy.

Comparisons with Different Sparsity Level

Table 3 shows our model's performance at different levels of sparsity by conducting experiments with varying numbers of sample points: 500, 200 and 50 in comparison to other existing methods, where the bold text indicates the best performance and the underlined text indicates the second-best performance. At 50 sample points, our model

shows the best performance with an RMSE of 0.270 and a REL of 0.058, demonstrating its effectiveness at lower levels of sparsity.

TABLE III. Comparisons with other methods

Method	Samples	RMSE↓	REL↓
pNCNN **Error! Reference source not found.**	500		
NLSPN **Error! Reference source not found.**		0.170	0.026
SpAgNet **Error! Reference source not found.**		<u>0.101</u>	**0.013**
SparseFormer **Error! Reference source not found.**		0.114	0.015
Ours		**0.104**	<u>0.014</u>
		0.206	0.052
pNCNN **Error! Reference source not found.**	200		
NLSPN **Error! Reference source not found.**		0.237	0.040
SpAgNet **Error! Reference source not found.**		**0.142**	**0.019**
SparseFormer **Error! Reference source not found.**		<u>0.155</u>	0.024
Ours		**0.142**	<u>0.022</u>
		0.227	0.056
pNCNN **Error! Reference source not found.**	50		
NLSPN **Error! Reference source not found.**		0.568	0.108
SpAgNet **Error! Reference source not found.**		0.423	<u>0.081</u>
SparseFormer **Error! Reference source not found.**		<u>0.272</u>	**0.058**
Ours		**-**	**-**
		0.270	**0.058**

Visual Comparison of Predicted Depth Maps

To provide a visual comparison of the predicted depth maps, Fig. 7 shows the depth completion results with different sampling strategies at 50 sample points. The results illustrate the qualitative performance of our model in handling sparse depth inputs.

Fig. 6. Depth completion visual results

V. CONCLUSION

In this paper, we proposed a depth completion network specifically designed for efficient and accurate depth map reconstruction. The proposed network leverages a dual-path autoencoder architecture with RGB and depth feature extractors, effectively integrating features from both RGB images and sparse depth inputs. Key components of the network, such as the scale and place (S&P) block, depth fusion (DF) block, and guided attention (G-Attn) mechanism, were evaluated through comprehensive experiments to demonstrate their contributions to overall performance. The experiments showed that the combination of these components significantly improves the accuracy of the depth completion model.

ACKNOWLEDGEMENT

This work was supported by Taiwanese National Science and Technology Council under Grant NSTC 113-2221-E-167-048-.

REFERENCES

[1] H. Hirschmuller, “Stereo processing by semiglobal matching and mutual information,” IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 328-341, 2007
[2] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: theory and experiment,” IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 920-932, 1994.
[3] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” Advances in Neural Information Processing Systems, pp. 2366–2374, 2014
[4] C. Godard, O. M. Aodha, G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 270–279, 2017
[5] Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, “Depth completion from sparse lidar data with depth-normal constraints,” in ICCV, 2019.
[6] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, “Learning guided convolutional network for depth completion. In IEEE TIP, pages 1116–1129, 2020.
[7] Park, J., Joo, K., Hu, Z., Liu, C.K., I. S Kweon, “Non-local spatial propagation network for depth completion,” in: ECCV, 2020
[8] F. Ma, G. V. Cavalheiro, and S. Karaman, “Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera,” Proc. of IEEE International Conference on Robotics and Automation, pp. 3288–3295. 2019.
[9] Y. Zhang and T. Funkhouser, “Deep depth completion of a single rgb-d image,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp.175–185, 2018.
[10] F. Warburg, M. Ramamonjisoa and M. López-Antequera, “Sparseformer: Attention-based depth completion network,” arXiv preprint arXiv:2206.04557, 2022.
[11] S. Liu, S. De Mello, J. Gu, G. Zhong, M.-H. Yang, and J. Kautz, “Learning affinity via spatial propagation networks,” Advances in Neural Information Processing Systems, vol. 30, 2017.
[12] X. Cheng, P. Wang, and R. Yang, ``Learning depth with convolutional spatial propagation network," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 10, pp. 2361-2379, 2019.
[13] K. Rho, J. Ha, and Y. Kim, “Guideformer: Transformers for image guided depth completion,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp. 6250–6259, 2022.

[14] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, " Swin transformer: Hierarchical vision transformer using shifted windows," Proc. of the IEEE International Conference on Computer Vision, pp. 10012–10022, 2021
[15] H. Tsai, J.-F. Yang. Low Resolution to High Precision Depth Estimation for MR Glasses, Master Thesis, NCKU 2023
[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.
[17] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, and S. Mattoccia, "Completionformer: Depth completion with convolutions and vision transformers. Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pp 18527–18536, 2023.
[18] A. Conti, M. Poggi and S. Mattoccia, "Sparsity Agnostic Depth Completion," 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2023, pp. 5860-5869
[19] J. Tang, F.-P. Tian, B. An, J. Li, P. Tan, Bilateral propagation network for depth completion, Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2024
[20] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," Proc. of European Conference on Computer Vision, pp. 746–760, 2012
[21] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," arXiv preprint arXiv:1711.05101, 2017.
[22] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end," Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12011–12020, 2020