electronics

MDPI

Article

Dense Feature Pyramid Deep Completion Network

Xiaoping Yang [1,2,3,*,†], Ping Ni [2,3], Zhenhua Li [1,*,†] and Guanghui Liu [4]

[1] Department of Information Physics and Engineering, School of Physics, Nanjing University of Science and Technology, Nanjing 210094, China
[2] College of Physics and Electronic Information Engineering, Guilin University of Technology, Guilin 541006, China; nip5077110@126.com
[3] Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, Guilin 541004, China
[4] Guilin Saipu Electronic Technology Limited Company, Guilin 541004, China; lgh9166@163.com
* Correspondence: gutyxp@126.com (X.Y.); lizhenhua@njust.edu.cn (Z.L.)
† These authors contributed equally to this work.

Abstract: Most current point cloud super-resolution reconstruction requires huge calculations and has low accuracy when facing large outdoor scenes; a Dense Feature Pyramid Network (DenseFPNet) is proposed for the feature-level fusion of images with low-resolution point clouds to generate higher-resolution point clouds, which can be utilized to solve the problem of the super-resolution reconstruction of 3D point clouds by turning it into a 2D depth map complementation problem, which can reduce the time and complexity of obtaining high-resolution point clouds only by LiDAR. The network first utilizes an image-guided feature extraction network based on RGBD-DenseNet as an encoder to extract multi-scale features, followed by an upsampling block as a decoder to gradually recover the size and details of the feature map. Additionally, the network connects the corresponding layers of the encoder and decoder through pyramid connections. Finally, experiments are conducted on the KITTI deep complementation dataset, and the network performs well in various metrics compared to other networks. It improves the RMSE by 17.71%, 16.60%, 7.11%, and 4.68% compared to the CSPD, Spade-RGBsD, Sparse-to-Dense, and GAENET.

Keywords: deep learning; depth complementation

Citation: Yang, X.; Ni, P.; Li, Z.; Liu, G. Dense Feature Pyramid Deep Completion Network. *Electronics* **2024**, *13*, 3490. https://doi.org/10.3390/electronics13173490

Academic Editors: Chih-Lung Lin and Chi-hung Chuang

Received: 24 June 2024
Revised: 23 August 2024
Accepted: 27 August 2024
Published: 2 September 2024

1. Introduction

With the advancement of technologies such as computer vision and 3D reconstruction [1], 3D data has experienced rapid development in fields such as virtual reality [2] and autonomous driving [3]. A point cloud, a common form of 3D data, constitutes a dataset composed of discrete points in 3D space, each carrying positional and other attribute information. It is the preferred choice for applications in scenarios such as autonomous driving and robotics analysis. The density of LiDAR point clouds involves the entire chain of LiDAR technology, including hardware manufacturing, data acquisition, and data processing and application, and is a key indicator of LiDAR technology. However, due to limitations of LiDAR sensors themselves, sampling intervals, or other factors, the acquired point cloud data may suffer from issues such as low resolution, loss of detail, or noise.

To address this issue, point cloud super-resolution techniques have been proposed, aiming to enhance the resolution and level of detail in point cloud data through algorithms and technologies. Traditional methods for point cloud super-resolution reconstruction primarily involve approaches such as bilateral filters [4,5], second-order generalized total variation [6,7], Markov random fields [8,9], and point cloud upsampling [10,11]. In recent years, with the advancement of deep learning, efforts towards super-resolution reconstruction based on deep learning [12–15] have also made significant progress.

In point cloud super-resolution reconstruction, low-resolution point clouds can be regarded as partially incomplete point clouds. Point cloud completion methods are then employed to fill in missing details and structures, thus achieving the goal of super-resolution reconstruction. In recent years, several outstanding deep learning-based sparse depth completion frameworks [16] have been proposed.

The convolutional spatial propagation network (CSPN) [17] is a deep learning architecture proposed in 2019 for the task of image inpainting, aiming to address the challenges of structural awareness and contextual consistency in image completion tasks. The network fills in missing regions in input images using convolutional and spatial propagation operations, making them appear more natural and coherent. The design objective of the CSPN is to integrate global and local contextual information in completion tasks to obtain more accurate and coherent completion results. Through spatial propagation and feature fusion mechanisms, the CSPN effectively utilizes contextual information in images to improve the quality of completion results. However, due to the localized nature of spatial propagation, it may not efficiently capture information from distant regions, leading to suboptimal performance when filling in large-scale missing areas.

Spade-RGBsD [18] is a jointly trained network architecture designed to integrate sparse depth (sD) data and dense RGB images for depth completion and semantic segmentation. It utilizes convolutional neural networks to process sparse depth data and dense RGB images, achieving depth completion and semantic segmentation tasks through joint training. By combining feature extraction, a depth completion module, and a semantic segmentation module, the network effectively integrates and processes sparse and dense data, thereby improving the accuracy of depth completion and semantic segmentation. However, in this network, only CNNs are used for the feature extraction and completion of sparse depth data, which may result in suboptimal performance when dealing with large-scale depth completion and semantic segmentation tasks.

Sparse-to-Dense [19] is a network architecture designed for depth completion, notable for its ability to generate dense depth maps from sparse depth data, thereby providing richer scene geometry information. By integrating features from sparse depth data and RGB images, it employs CNNs for feature extraction and fusion, generating high-resolution depth maps through upsampling and convolution operations. The network's training process enables it to learn from a large amount of annotated depth data, utilizing the difference between annotated dense depth maps and network-generated depth maps to train the network. Supervised learning is employed using the backpropagation algorithm to optimize depth prediction accuracy. The approach to the loss function is noteworthy, but the model exhibits relatively weak learning capabilities and slow convergence speeds.

The Geometry-Aware Embedding Network (GAENet) [20] is a simple yet effective depth completion method that integrates 3D geometry representations into a 2D learning architecture, achieving a better balance between performance and efficiency. This paper proposes an efficient geometry-aware embedding learning approach that encodes local and global geometric structural information about 3D points such as scene layout, object sizes, and shapes to guide dense depth estimation. Combining this embedding with corresponding RGB appearance information allows for the inference of missing depths while preserving well-structured details of the scene. The key of this method lies in integrating an implicit 3D geometry representation into a 2D learning architecture, thus achieving a better balance between performance and efficiency. However, this network requires mapping the depth map to a 3D point cloud for edge convolution to extract object edge information, which increases the computational complexity of the network and may add to the training time cost.

This paper proposes a dense feature pyramid depth completion network for generating higher-resolution point clouds by fusing features from images and low-resolution point clouds at the feature level. The network is an improvement upon a feature pyramid-based architecture, utilizing an image-guided feature extraction network, RGBD-DenseNet, as the encoder to extract multi-scale features. Subsequently, an upsampling block formed by

operations such as transpose convolution serves as the decoder to gradually restore the size and details of the feature maps. Additionally, pyramid connections are employed to connect corresponding levels of the encoder and decoder. To further enhance network performance, a joint loss function comprising smoothness loss and Laplacian pyramid loss is proposed. By utilizing this network framework and joint loss function, image information serves as guidance, enabling sparse depth maps to undergo multi-scale feature extraction and fusion to output dense depth maps. This network presents a novel end-to-end solution for depth completion tasks.

The following outlines our upcoming work: In Section 2, we describe further details of DenseFPNet, including the specific network architecture and parameters. Section 3 primarily presents the experimental results, including dataset processing, loss function, environment configuration, and detailed analysis. Finally, Section 4 provides an overview of the work completed and the results achieved.

2. Methodology

2.1. Dense Networks DenseNet and Its Variants

ResNet (Residual Neural Network) [21] is a deep convolutional neural network model proposed by the Microsoft Research team in 2015. The main idea is to address issues such as gradient vanishing and model degradation during the training process by introducing residual connections and constructing residual blocks. This connection method helps the network more effectively learn and capture features from the input data. DenseNet (densely connected convolutional network) is a further development based on ResNet, proposed by Huang et al. in 2017 [22]; it introduces dense connections to construct dense blocks, connecting all preceding layers with subsequent layers densely. This connection method achieves feature reuse across channels, enabling DenseNet to outperform ResNet with fewer parameters and lower computational costs.

In DenseNet, each layer's output is concatenated with the inputs of all subsequent layers, forming dense blocks. Each dense block consists of several bottleneck convolutional layers, arranged as follows: BN (batch normalization), ReLU (activation function), 1×1 convolution, BN, and ReLU, 3×3 convolution. This structure fully utilizes the feature information from preceding layers and performs feature transformation and integration through activation functions and convolutional layers. Additionally, transition layers are included between adjacent dense blocks, composed of BN, ReLU, 1×1 convolution, and 2×2 average pooling operations. The 1×1 convolution is used to reduce feature dimensions, thereby decreasing the model's parameter count, while average pooling halves the size of the feature maps. Through transition layers, DenseNet can control the number and size of feature maps, further reducing the parameter count and improving network efficiency. This design enables DenseNet to achieve higher parameter efficiency, computational efficiency, and better utilization of feature information, leading to improved performance.

DenseNet has four variants, DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264, differing mainly in network depth and parameter count.

DenseNet-121: DenseNet-121 is the most basic variant of DenseNet, consisting of 121 layers. It has a relatively shallow network structure and fewer parameters, making it suitable for small-scale datasets or scenarios with limited computational resources.

DenseNet-169: DenseNet-169 contains more layers than DenseNet-121, totaling 169 layers. Compared to DenseNet-121, DenseNet-169 has a deeper network structure and more parameters, allowing it to more effectively capture details and complex features in images. It is suitable for medium-scale datasets and tasks.

DenseNet-201: DenseNet-201 further increases the depth and parameter count of the network, comprising 201 layers. Compared to DenseNet-169, DenseNet-201 deepens the network structure further, providing stronger feature extraction and representation capabilities. It is applicable to larger-scale datasets and more complex tasks, but may require more training samples, a longer training time, and more sophisticated optimization strategies to achieve optimal performance.

DenseNet-264: DenseNet-264 is the deepest variant in the DenseNet series, containing 264 layers. It has the maximum network depth and the highest number of parameters, but also requires more computational resources and a longer training time.

Overall, DenseNet-121, DenseNet-169, DenseNet-201, and DenseNet-264 differ in network depth and parameter count. With increasing network depth, these variants can provide more powerful feature representation capabilities but also increase computational and storage requirements.

2.2. Network Architecture

By analyzing the shortcomings of existing depth completion network frameworks, this paper proposes an improvement based on the pyramid network architecture, introducing a novel Dense Feature Pyramid Network (DenseFPNet) derived from an enhanced DenseNet. DenseFPNet integrates RGB image data and sparse depth maps, and through multi-scale feature extraction and fusion, it outputs dense depth maps. The architecture of DenseFPNet consists of three main components, an encoder, a decoder, and pyramid connections, as illustrated in Figure 1.

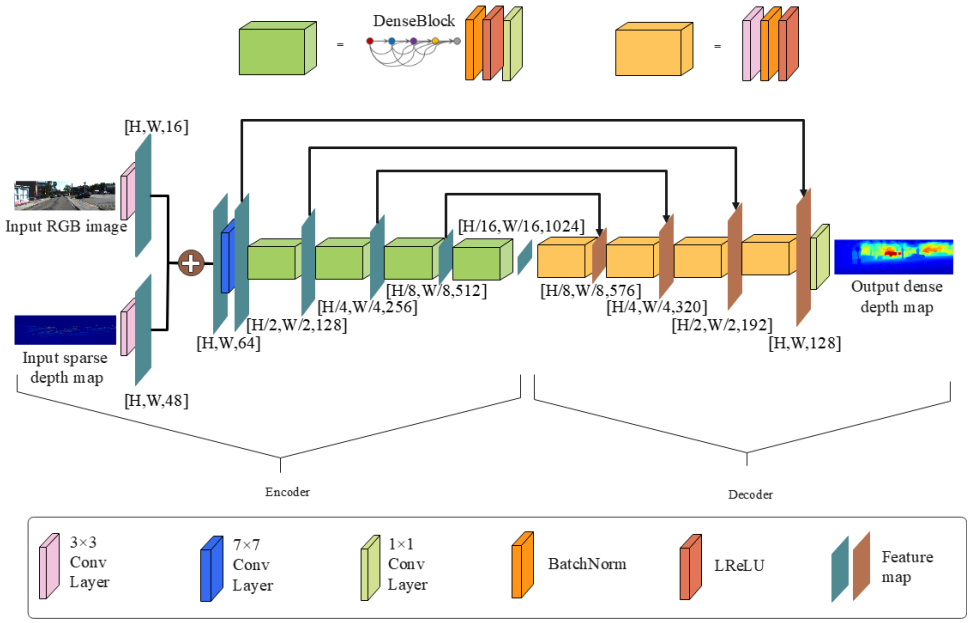


Figure 1. DenseFPNet.

The encoder section consists of multiple convolutional and pooling layers, responsible for processing the input RGB image data and sparse depth maps. It conducts feature fusion and multi-scale feature extraction to abstract both low-level and high-level features from the input data, encoding them into higher-level representations for decoder utilization. The decoder section comprises multiple upsampling blocks and operations for feature fusion, gradually restoring the size and feature details of the feature maps. It decodes the feature maps extracted by the encoder into outputs of the same size as the original input data, ultimately generating predictions or reconstructions matching the original input data.

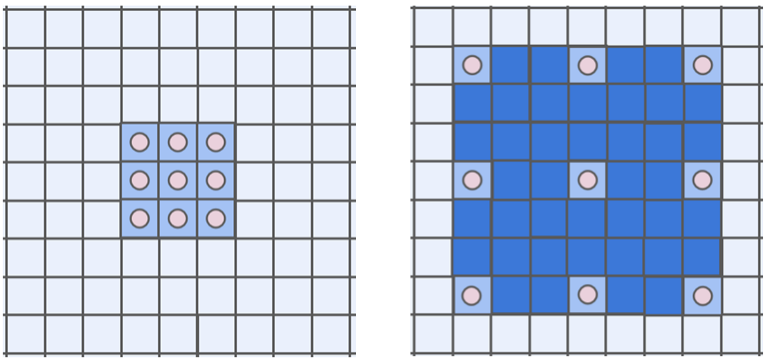
Pyramid connections exist between the encoder and decoder for feature propagation and information fusion. Pyramid connections represent a hierarchical connectivity approach, linking corresponding levels of the encoder and decoder to form a pyramid-like connectivity pattern. Features from each level are utilized in the corresponding level of the decoder to fuse information from different scales. In this network framework, the purpose of pyramid connections is to introduce multi-scale information into the decoder, thereby enhancing its modeling capabilities for features of varying scales.

2.2.1. Encoder

Among the four variants of DenseNet introduced, DenseNet-169, compared to DenseNet-121, features more parameters and a more complex structure. It can handle more images and provide stronger expressive capabilities while conserving computational resources and training time compared to other variants. Therefore, this paper focuses on improving the task of depth map completion through the fusion of images and sparse depth maps, building upon DenseNet-169. It proposes a feature extraction network named RGBD-DenseNet, which serves as the encoder part of DenseFPNet for multi-scale feature extraction and abstraction.

RGBD-DenseNet similarly employs four dense blocks and transition layers for feature extraction and downsampling operations, retaining the dense connectivity architecture within dense blocks to address gradient vanishing issues. The improvements over DenseNet-169 are as follows:

1. Before conducting various convolution operations, the feature extraction and fusion of the input RGB image and depth map information are required. This paper utilizes 3×3 convolutional layers to map them into 16-channel and 48-channel feature maps, which are then merged into a 64-channel joint feature map.
2. The joint feature map is merely a consolidation of channel numbers; thus, before conducting multi-scale feature extraction, further convolution operations are required for feature extraction. In this study, we employ 3×3 dilated convolutions to extract features from the joint feature map, with a dilation rate of 3, a stride of 1, and the channel number remaining unchanged. As illustrated in Figure 2, compared to regular convolutions, the dilated convolutions utilized in this study offer a larger receptive field, enabling the extraction of more features. While maintaining the size of the feature map, they gather a broader range of contextual information, facilitating more accurate feature extraction and prediction.
3. The average pooling in the transition layers is removed, retaining only batch normalization, ReLU, and 1×1 convolution. Pooling operations lead to information loss and the compression of feature map size, and their removal ensures the size of the feature map remains unchanged while preserving more details.
4. A dense block is combined with its subsequent transition layer into a dense convolutional block. Additionally, the channel numbers of the feature maps obtained after four dense convolutional blocks are changed to 128, 256, 512, and 1024, respectively, and the feature map sizes are reduced to $1/2$, $1/4$, $1/8$, and $1/16$ of the original data. Through four-fold downsampling operations, features from each scale can be obtained as much as possible, while outliers and noise can also be removed.
5. The global average pooling layer, final fully connected layer, and softmax function are removed, and the feature maps obtained from the fourth dense block are directly connected to the decoder.



(a) (b)

Figure 2. Schematic comparison of regular convolution and dilated convolution used in this paper. (**a**) Dilation rate = 1, (**b**) Dilation rate = 3.

2.2.2. Decoder

The decoder of DenseFPNet comprises four upsampling blocks and one 1×1 convolution. As illustrated in Figure 3, each upsampling block consists of a 3×3 transpose convolution, a batch normalization layer, and an LReLU activation function layer, where the slope of each LReLU is set to -0.2. Additionally, pyramid connections are utilized to merge the features from corresponding levels of the encoder with the features obtained from the upsampling blocks, before being inputted together into the next upsampling block. This prevents the loss of original details from the sparse depth map during the upsampling process.

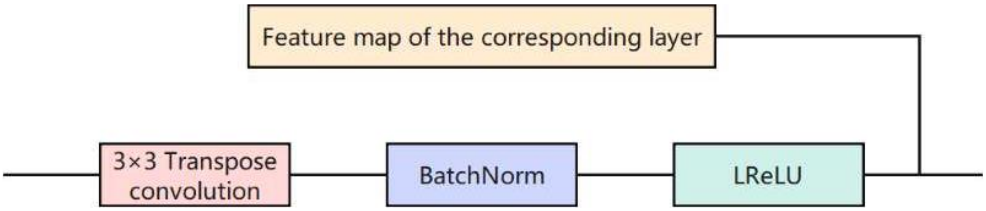


Figure 3. Schematic of the upsampling block in the decoder.

The 3×3 transpose convolution operates by convolving the input feature map with a transpose convolution kernel, allowing for the enlargement of low-resolution feature maps to higher resolutions, thus restoring spatial details lost during downsampling. Batch normalization, employed in the upsampling blocks, primarily normalizes the feature maps across each batch, standardizing the numerical range of the feature maps and enhancing the model's generalization capability. LReLU (Leaky ReLU) serves as a rectified linear unit activation function, introducing non-linearity to enable the network to learn more complex feature representations while maintaining computational efficiency. Compared to traditional ReLU, LReLU introduces a small negative slope in the negative region to mitigate the "dying neuron" phenomenon.

Each upsampling block outputs feature maps with 64 channels. Except for the first upsampling block, all subsequent modules require merging the number of channels from the output of the previous upsampling block with the corresponding output channels from the encoder. Thus, the input channels for each upsampling block are 1024, 576, 320, 192, and 128. Following each upsampling block, the size of the obtained feature maps is restored to twice their original size.

3. Results and Discussion

3.1. Datasets and Their Processing

To benchmark state-of-the-art methods, we utilized the KITTI Depth dataset [23] for training and testing. This dataset aggregates 11 consecutive LiDAR scans to generate approximately 30% annotated pixels of semi-dense ground truth. We employed the entire dataset comprising 85,898 training samples, 1000 selected validation samples, and 1000 test samples lacking ground truth, as shown in the example image in Figure 4.

To increase the diversity of the data and enhance the robustness and generalization capability of the model, random geometric and color transformations were applied to the input data during the training process. The operations used for training data in this paper are as follows:

1. Rotation: RGB images and depth images are rotated by a random rotation angle r within the range of $[-5, 5]$ degrees.
2. Horizontal flipping: RGB images and depth images are horizontally flipped with a probability $p = 0.5$.
3. Color distortion: Brightness, contrast, and saturation values are increased or decreased by multiplying them by a distortion factor k within the range of [0.5, 1.5].

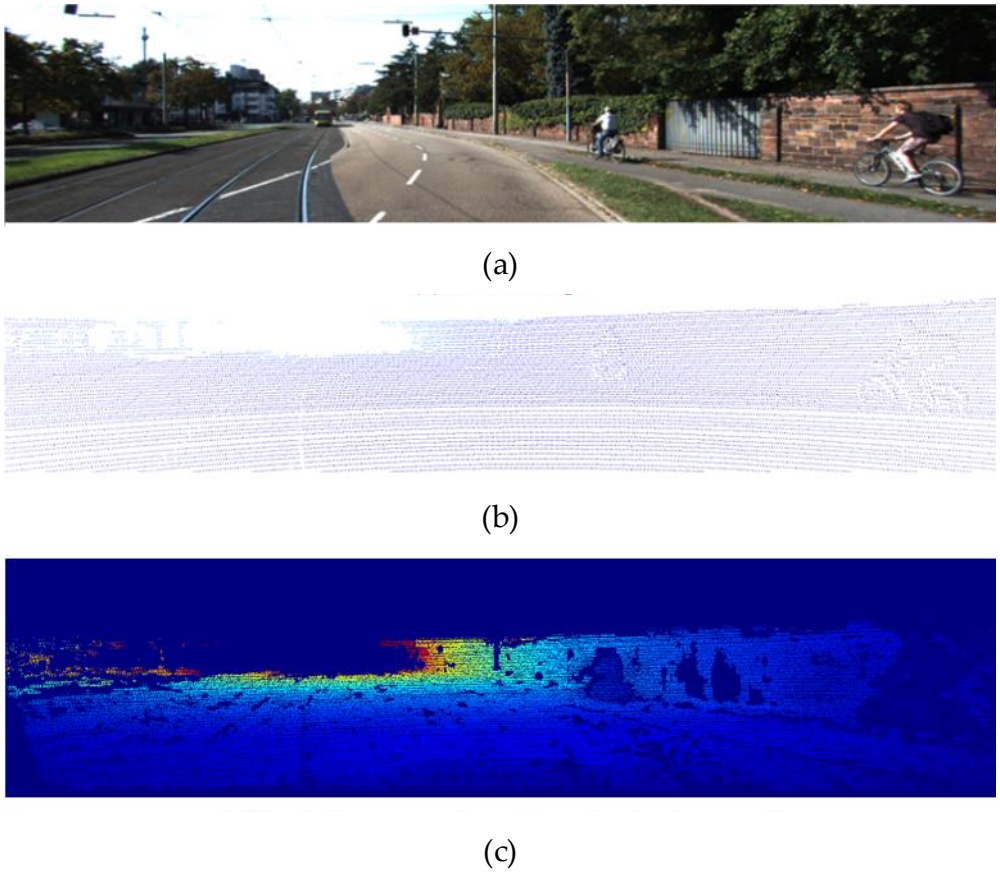


Figure 4. Example plot of KITTI dataset. (**a**) RGB Image, (**b**) Sparse depth map (white background), (**c**) Semi-dense map (pseudo-color background).

These operations aim to introduce variations in the training data, helping the model to learn features that are invariant to such transformations and improving its ability to generalize to unseen data.

3.2. Loss Function

Huber Loss is a regression loss function, also known as Smooth L1 Loss. It is called "smooth" because it uses mean squared error loss when the difference is small and absolute error loss when the difference is large, thereby achieving a smooth transition.

The formula for Huber Loss is as follows:

$$X = \begin{cases} \frac{1}{2}(y - f(x))^2 & if\ |y - f(x)| \leq \sigma \\ \sigma|y - f(x)| - \frac{1}{2}\sigma & if\ |y - f(x)| > \sigma \end{cases} \tag{1}$$

In the equation, y represents the target value, $f(x)$ denotes the predicted value, and σ serves as a hyperparameter that controls the smoothness of the loss function between squared loss and absolute loss. A smaller σ value makes the loss function closer to squared loss, thus being more sensitive to outliers. Conversely, a larger σ value makes the loss function closer to absolute loss, rendering it more robust to outliers.

Laplacian Pyramid Loss is a loss function used for image generation and image style transfer tasks. It is based on the concept of Laplacian pyramids, measuring the difference between generated images and target images through multi-scale decomposition and reconstruction of images. Its mathematical expression is defined as follows:

$$L_{ap} = \sum 2^{i-1} \left|\left| L^i(\hat{\alpha}) - L^i(\alpha) \right|\right|_1 \tag{2}$$

where $L^i(\hat{\alpha})$ represents the result of the predicted α at the i-th level of the Laplacian pyramid.

A single loss function may not fully capture all crucial objectives or metrics, and in some cases, it may lead to training difficulties or susceptibility to overfitting. To address these issues, a joint loss function can integrate multiple objectives or metrics, enabling the model to optimize across multiple aspects. Additionally, it can provide more information and constraints, guiding the model towards better learning directions and effectively exploring the parameter space. The joint loss function proposed in this paper is depicted as follows in Equation (1):

$$L = L_H + L_{ap} \tag{3}$$

The proposed joint loss function in this paper enables the comprehensive consideration of multiple objectives, enhancing the model's robustness without sacrificing depth map details. It helps alleviate training difficulties and overfitting, while providing stronger optimization signals.

3.3. Environment Configuration and Parameter Setting

The network proposed in this paper is based on the fusion of image and sparse depth map data and utilizes deep learning techniques. Due to the high computational complexity of the required algorithms, a combination of CPU and GPU processing is employed in the experiments conducted in this chapter. The specific computing environment used is detailed in Table 1.

Table 1. Experimental environment.

System/Platform	Configuration/Version
CPU (Intel Corporation, Santa Clara, CA, USA)	12th GenIntel(R) Core(TM) i5-12400F
Memory	64G
GPU (NVIDIA Corporation, Santa Clara, CA, USA)	NVIDIA GeForce RTX3060 (16G)
OS	Ubuntu16.04
Programming Language	Python3.7
Deep Learning Framework	Pytorch1.13.1

In the experiments conducted in this paper, the network architectures were trained for 11 epochs. The initial learning rate was set to 0.001, and every five epochs, the learning rate was decayed by a factor of 0.1. To optimize the adjustment of the learning rate, the Adam optimizer was employed. This setup allowed the network to adaptively adjust the learning rate based on the training progress. The variation in the learning rate during the training of the proposed network architecture is depicted in Figure 5, which was automatically generated by the Tensorboard tool. In the graph, the horizontal axis represents the number of training data samples, while the vertical axis represents the learning rate.

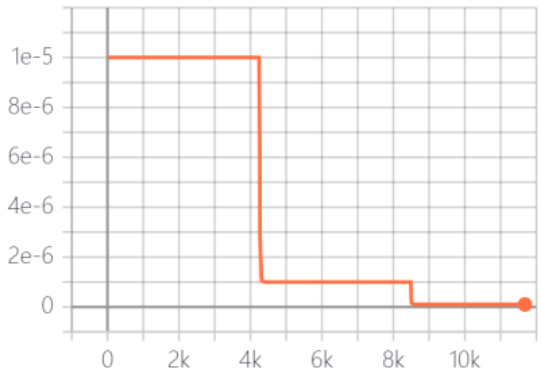


Figure 5. The graph of the learning rate change during the training of this network.

3.4. Analysis of Experimental Results

In summary, DenseFPNet modifies the DenseNet backbone network, including the corresponding layers of the encoder for multi-scale feature extraction and the pyramid architecture for the decoder. The encoder allows the network to capture information at various levels of detail, ranging from fine-grained features to broader contextual cues. The pyramid connections introduce these multi-scale features into the decoder, which is crucial for accurately representing object edges. Edges in images can vary significantly in size and prominence, and by analyzing features at different scales, DenseFPNet can better detect and delineate edges, ensuring that both small and large-scale features are accurately represented.

To evaluate the performance of the proposed network architecture DenseFPNet, extensive experiments were conducted in this study, including comparisons with other depth completion networks and ablation studies. To validate the effectiveness of the experiments, the same dataset was used for all experiments. As the proposed depth completion network architecture is primarily applied to the task of the super-resolution reconstruction of vehicle LiDAR point clouds in autonomous driving, the experiments were conducted on the outdoor depth completion dataset of KITTI.

The evaluation metrics for depth completion algorithms typically compare the valid depth values from the ground truth d_v^{gt} with the corresponding predicted depth values d_v^{pred}. Based on different distance measurement standards, these metrics can be categorized into four types: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), inverse RMSE (iRMSE), and inverse MAE (iMAE). RMSE is sensitive to large errors, thus giving more weight to larger prediction errors. MAE provides a direct measure of the error magnitude without considering the squared error, making it less sensitive to outliers. iRMSE assesses the error in the inverse depth values, emphasizing errors at small depth values, making it more sensitive to small depth errors. iMAE also focuses on the inverse depth errors, making it suitable for evaluating the prediction accuracy at smaller depth values. The formulas for these four evaluation metrics are as follows:

$$RMSE = \sqrt{\frac{1}{|V|}\sum_{v\in V}\left|d_v^{gt} - d_v^{pred}\right|^2} \tag{4}$$

$$MAE = \frac{1}{|V|}\sum_{v\in V}\left|d_v^{gt} - d_v^{pred}\right| \tag{5}$$

$$\mathrm{i}RMSE = \sqrt{\frac{1}{|V|}\sum_{v\in V}\left|\frac{1}{d_v^{gt}} - \frac{1}{d_v^{pred}}\right|^2} \tag{6}$$

$$iMAE = \frac{1}{|V|}\sum_{v\in V}\left|\frac{1}{d_v^{gt}} - \frac{1}{d_v^{pred}}\right| \tag{7}$$

3.4.1. Comparison with Other Networks on KITTI Dataset

The study compares the proposed DenseFPNet architecture for depth completion with four existing state-of-the-art depth completion networks to validate the effectiveness of the proposed model. Similar to the proposed model, these four models are also trained end-to-end in an autoencoder manner.

As shown in Table 2, the proposed network architecture reduces the RMSE, a key metric, to 756.75, improving by 17.71%, 16.60%, 7.11%, and 4.68% compared to the other four networks. The MAE is reduced to 223.15, achieving significant improvements over the other four methods. The primary reason for this might be that the compared networks did not discuss their performance under noisy or disturbed data conditions in their original papers, which may indicate they did not consider the ability to handle noise and disturbed data, resulting in poorer performance in such scenarios. On the other hand, DenseFPNet,

which employs a pyramid architecture, is able to minimize noise through multi-scale feature extraction, thereby reducing error accumulation caused by noise. However, the proposed network architecture only reduces the iRMSE to 2.21 and the iMAE to 1.13, with performance in these two metrics falling short of Spade-RGBsD. This may be due to the use of a smoothness loss function in our model, leading to over-smoothing in some areas. Nonetheless, it still outperforms other methods. Overall, the network architecture proposed in this paper demonstrates better and more stable performance compared to the other four networks.

As shown in Figure 6, three examples are selected for comparison with each network, with distinct differences highlighted by yellow dashed boxes. The CSPN and Spade-RGBsD methods fail to adequately display details of objects slightly further away. Particularly evident in the third example, the edges of the car approaching from a distance are blurred in the depth maps generated by these two networks, making it difficult to discern the outline of the car. In contrast, the proposed network architecture renders the edges and contours of the car clearer, indicating better handling of details of distant objects. Sparse-to-Dense and GAENET excessively smooth the edges and details of objects. This is particularly noticeable in the second example, where the truck's edges blend into the surrounding scene, obscuring the boundaries between the truck and the environment. In contrast, the proposed network architecture exhibits clearer delineation of the truck's edges.

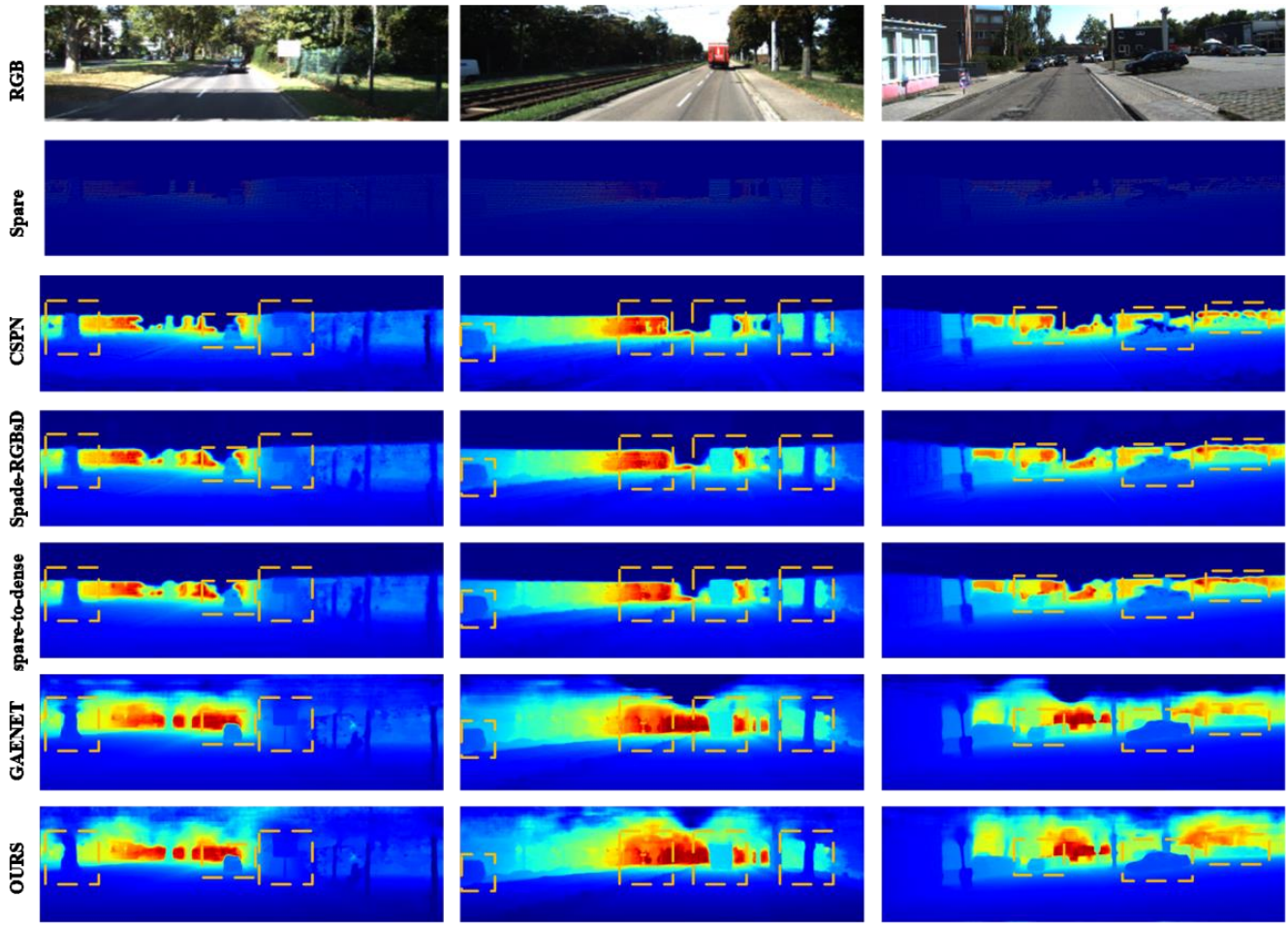


Figure 6. Visual comparison with other networks for different examples.

In summary, the proposed network architecture demonstrates superior performance in qualitative comparison experiments, clearly displaying object edges both near and far.

Table 2. Quantitative evaluation of upsampled networks. Table of quantitative experimental results.

Network	RMSE	MAE	iRMSE	iMAE
CSPN	919.64	279.46	2.63	1.25
Spade-RGBsD	907.34	234.81	**2.17**	**0.95**
Sparse-to-Dense	814.73	249.95	2.8	1.2.1
GAENET	793.90	231.29	2.27	1.08
Ours	**756.75**	**223.15**	2.21	1.13

As shown in Figure 7, detailed comparisons were conducted between the proposed method and CSPN and Sparse-to-Dense on the same example. For each sample, the input RGB image, the input sparse depth map, the depth map completed by each network, and some details of the completed depth maps after magnification are presented. A comparison of the magnified details reveals that the proposed method produces depth maps with clearer object edges and higher discernibility.

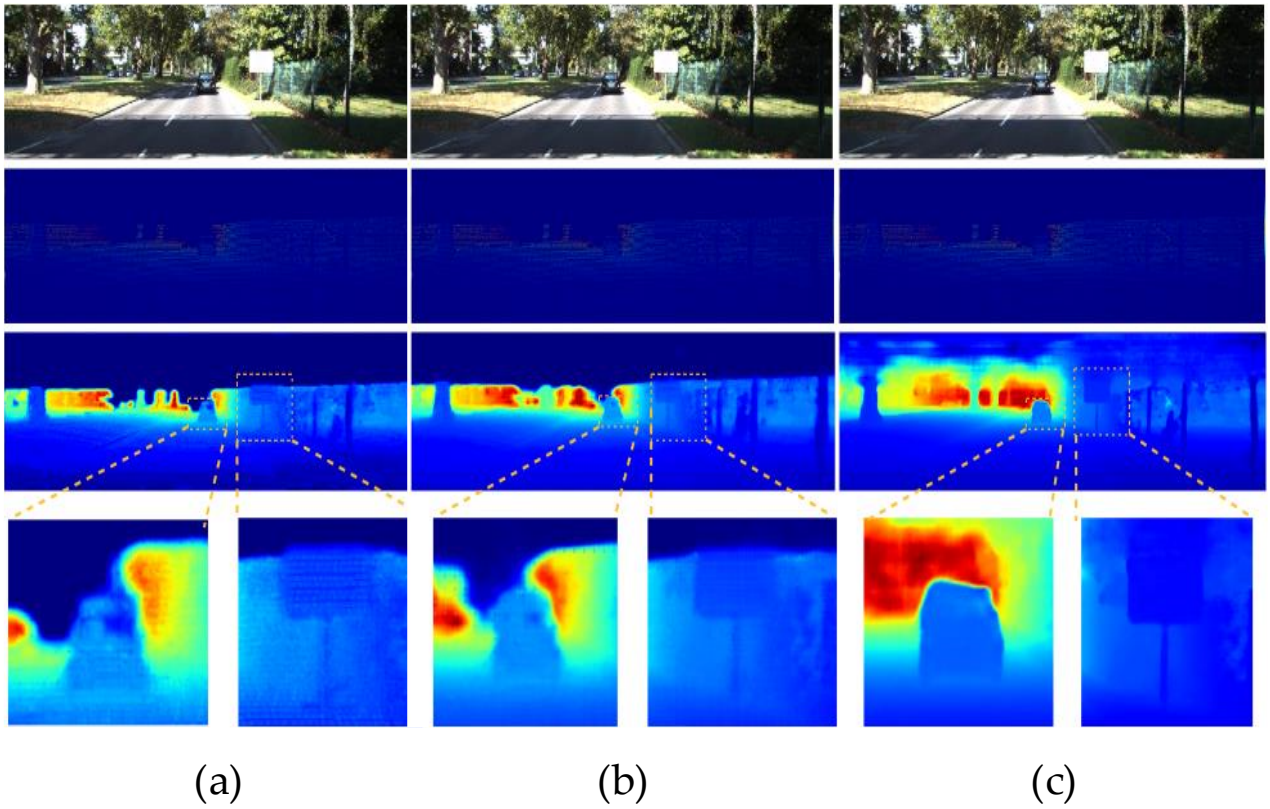


Figure 7. Detailed comparison with other networks under the same example. (**a**) CSPN, (**b**) Sparse-to-Dense, (**c**) ours.

Overall, the proposed method generates denser depth maps that are more accurate and exhibit better details.

3.4.2. Ablation Experiment

The loss functions used in this study consist of two components: L_H and L_{ap}. To verify the impact of these two loss functions on the final completed depth maps, two separate depth completion models were trained: one using only L_H loss and the other using only L_{ap} loss. Both models were based on the proposed DenseFPNet architecture. The comparison of the loss function values during training for these two models and the proposed model is shown in Figure 8. This figure, automatically generated by the Tensorboard tool during

training, illustrates the training data quantity on the x-axis and the values of the respective loss functions on the y-axis.

As shown in Table 3, this study conducted a comparison of four evaluation metrics in the ablation experiments. The results indicate that removing any single loss function leads to inferior performance in each evaluation metric compared to the joint loss. This observation suggests that using both loss functions together in the network architecture yields the best results.

Table 3. Quantitative evaluation of upsampled networks. Table of quantitative comparison of ablation experiments.

Loss Function	RMSE	MAE	iRMSE	iMAE
L_H	808.63	263.98	2.74	1.2.5
L_{ap}	786.50	254.69	2.59	1.18
Ours (L_H + L_{ap})	**756.75**	**223.15**	**2.27**	**1.13**

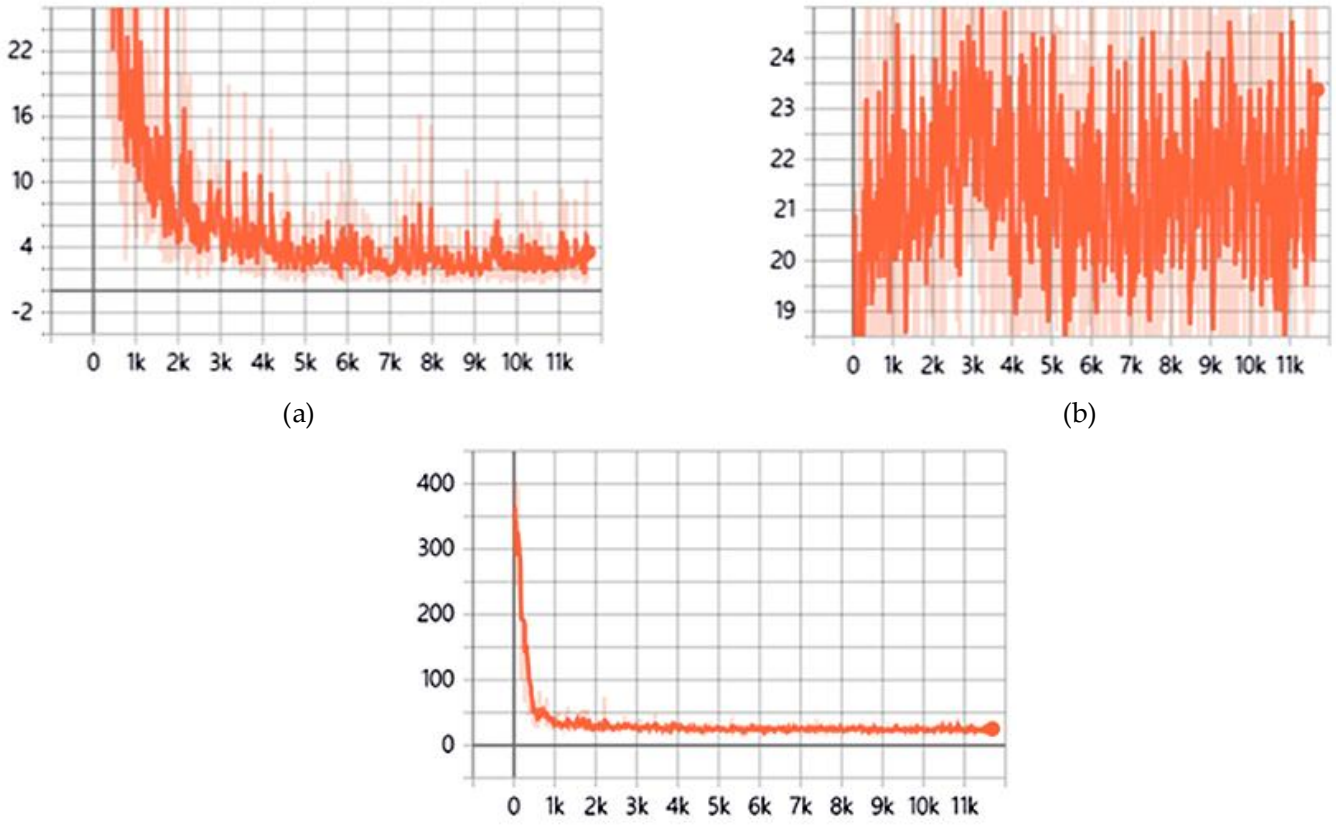


Figure 8. Comparison of loss function values of three network models. (**a**) L_H, (**b**) L_{ap}, (**c**) L_H + L_{ap}.

As shown in Figure 9, this study conducted a qualitative visualization comparison with LH and Lap in a single sample, highlighting the differences in more detail with yellow dashed-line boxes. Compared to the depth map obtained with the joint loss, the depth map trained solely with LH loss exhibited excessive smoothing effects in areas such as car windows and overlapping vehicles. On the other hand, the depth map trained solely with Lap loss showed severe ghosting issues in the overlapping parts of vehicles. Therefore, in the proposed model of this study, removing either loss function did not lead to better depth completion, highlighting the superiority of the proposed joint loss function.

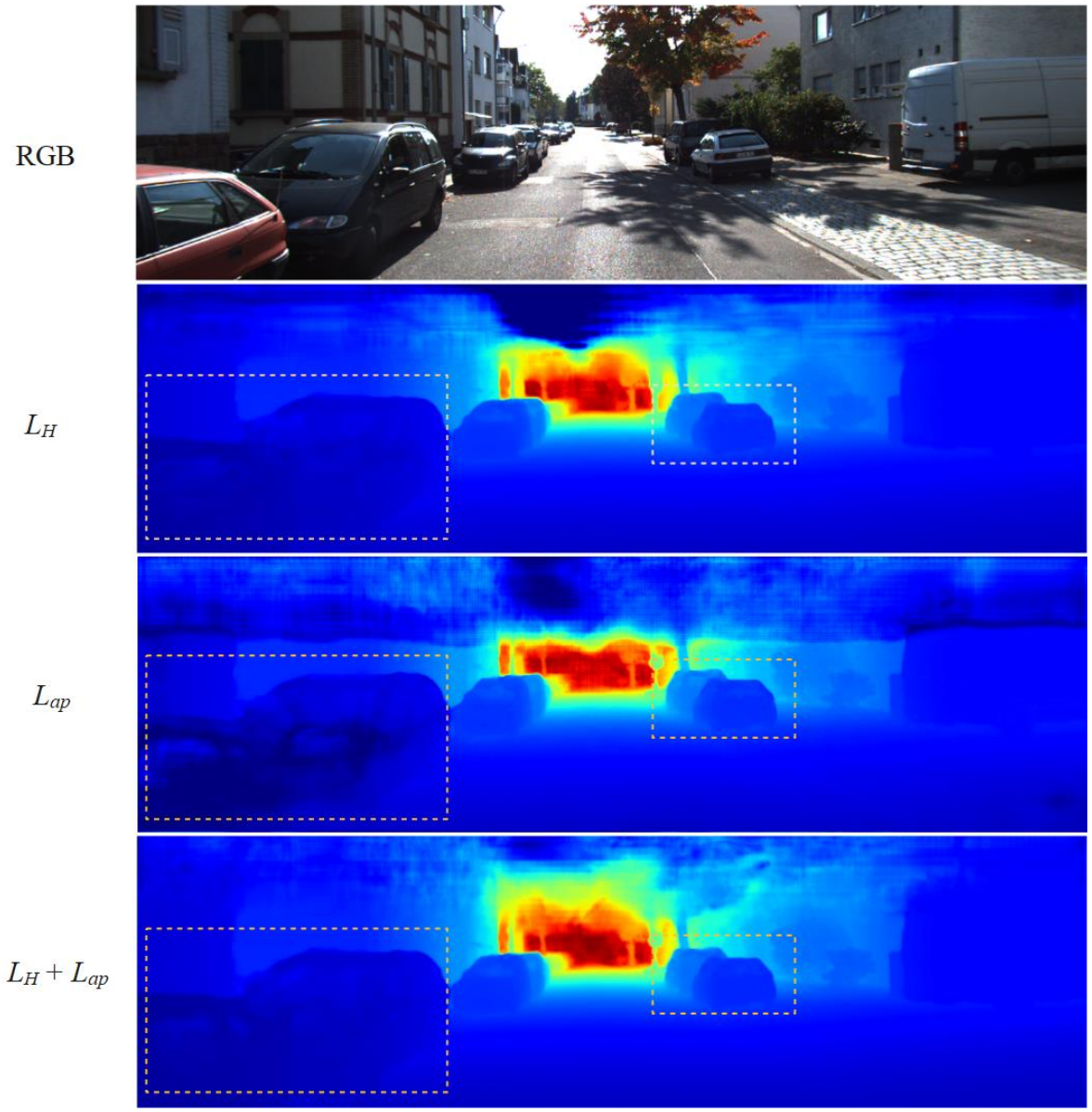


Figure 9. Visualization rendering of qualitative comparison of ablation experiments.

4. Conclusions

This paper addresses issues present in existing depth completion network frameworks, including filling large-scale depth gaps, weak learning capacity, and insufficient detail representation. To tackle these challenges, the paper proposes DenseFPNet, a dense feature pyramid depth completion network. Comparative experiments with other depth completion networks demonstrate that DenseFPNet achieves superior performance, reducing the RMSE metric to 756.75. Compared to the other four networks, this represents improvements of 17.71%, 16.60%, 7.11%, and 4.68%, respectively. Furthermore, to further enhance network performance, a joint loss function is proposed, comprising smoothness loss and Laplacian pyramid loss, with ablation experiments confirming the superiority of this joint loss function.

Author Contributions: X.Y.: Conceptualization; Methodology; Project Administration; Formal Analysis; Writing—Original Draft; P.N.: Software; Investigation; Resources; Z.L.: Writing—Review and Editing; Investigation; Supervision; Validation; G.L.: Data Curation; Visualization; Writing—Review and Editing. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the National Natural Foundation of China (NSFC62221004); Guangxi Science and Technology Major Program (AA23023027); Guangxi Key Research and Development Program (AB23026149, AB24010073); Guilin Scientific Research Project (20220113-1); Guilin Scientific Research Project (20220107-1); Guilin Scientific Research Project (20210101-5).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Author Guanghui Liu was employed by the Guilin Saipu Electronic Technology Limited Company. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Chen, S. Research on depth Completion Algorithm Based on fusion of lidar and camera. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2022.
2. Xiu, Y.L.; Yang, J.L.; Tzionas, D.; Black, M.J. ICON: Implicit clothed humans obtained from normals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13286–13296.
3. Yin, T.W.; Zhou, X.Y.; Krahenbuhl, P. Center-based 3D object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11779–11788.
4. Tu, Y.; Zhang, X.; Zhang, J.; Hu, L. Depth Image super-resolution Reconstruction Guided by Edge features. *Comput. Appl. Softw.* **2017**, *34*, 220–225. [CrossRef]
5. He, K.M.; Sun, J.; Tang, X.O. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. [CrossRef] [PubMed]
6. Di, W.; Zhang, X.; Hu, L.; Duan, L. Second-order Total generalized variation for depth Map super-resolution Reconstruction Constrained by Color images. *J. Image Graph.* **2014**, *19*, 1162–1167. [CrossRef]
7. Wang, Y.; Pu, Y.; Sun, R. Depth map super-resolution reconstruction combined with color image of the same scene. *Acta Opt. Sin.* **2017**, *37*, 810002. [CrossRef]
8. Chen, J.; Li, R. A Depth Map Super-Resolution Reconstruction Based on Improved MRF. *Microprocessors* **2017**, *38*, 60–63, 71.
9. Park, J.; Kim, H.; Tai, Y.W.; Brown, M.S.; Kweon, I. High quality depth map upsampling for 3D-TOF cameras. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 1623–1630.
10. Rock, J.; Gupta, T.; Thorsen, J.; Gwak, J.Y.; Shin, D.; Hoiem, D. Completing 3D object shape from one depth image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2484–2493.
11. Qian, G.C.; Abualshour, A.; Li, G.H.; Thabet, A.; Ghanem, B. PU-GCN: Point cloud upsampling using graph convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 11678–11687.
12. Achlioptas, P.; Diamanti, O.; Mitliagkas, I.; Guibas, L. Learning representations and generative models for 3d point clouds. In Proceedings of the International Conference on Machine Learning, Macau, China, 26–28 February 2018; pp. 40–49.
13. Yang, Y.Q.; Feng, C.; Shen, Y.R.; Tian, D. FoldingNet: Point cloud auto-encoder via deep grid deformation. *arXiv* **2017**, arXiv:1712.07262.
14. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. TopNet: Structural point cloud Decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 383–392.
15. Huang, Z.T.; Yu, Y.K.; Xu, J.W.; Ni, F.; Le, X. PF-net: Point fractal network for 3D point cloud completion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7659–7667.
16. Hu, J.; Bao, C.; Ozay, M.; Fan, C.; Gao, Q.; Liu, H.; Lam, T.L. Deep depth completion from extremely sparse data: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 8244–8264. [CrossRef] [PubMed]
17. Cheng, X.; Wang, P.; Yang, R. Depth estimation via affinity learned with convolutional spatial propagation network. In Proceedings of the European Conference on Computer Vision (ECCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 103–119.
18. Jaritz, M.; Charette, D.R.; Wirbe, E.; Perrotton, X.; Nashashibi, F. Sparse and dense data with cnns: Depth completion and semantic segmentation. In Proceedings of the International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 52–60.
19. Ma, F.; Cavalheiro, G.V.; Karaman, S. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 3288–3295.
20. Du, W.C.; Chen, H.; Yang, H.Y.; Zhang, Y. Depth Completion using Geometry-Aware Embedding. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.
21. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

22. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
23. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]