T-UNet: A Novel TC-Based Point Cloud Super-Resolution Model for Mechanical LiDAR

Lu Ren[2,3,4], Deyi Li[2,3,4], Zhenchao Ouyang[1,3(✉)], Jianwei Niu[2,3], and Wen He[4,5]

[1] Zhongfa Aviation University, Hangzhou 310000, Zhejiang, China
[2] State Key Laboratory of Software Development Environment, BeiHang University, Beijing 100191, China
[3] Hangzhou Innovation Institute, BeiHang University, Hangzhou 310000, Zhejiang, China
[4] Nanhu Laboratory, Jiaxin 314000, Zhejiang, China
ouyangkid@buaa.edu.cn
[5] Chinese Academy of Military Science, Beijing, China

Abstract. Mechanical LiDAR is one of the most crucial perception sensors for autonomous vehicles. However, the vertical angular resolution of low-cost multi-beam LiDAR is small, limiting the perception and movement range of mobile agents. This paper presents a novel temporal convolutional (TC)-based U-Net model for point cloud super-resolution, which can optimize the point cloud of low-cost LiDAR based on fusing spatiotemporal features of the point cloud. We project the 3D point cloud on a 2D image plane and extend a U-Net convolutional neural network model with a temporal convolutional (TC) module for processing consecutive frames. Each time the model generates one dense/up-sampled image from low-end LiDAR consecutive frames and projects it back into the 3D space as the final result. Considering the intrinsic noise of LiDAR, the structural similarity index measure (SSIM) is introduced as the loss function. Experiments are carried out on both datasets generated by the CARLA simulator and a small-scale dataset collected from actual road conditions with a local vehicle platform. Results show that the proposed model achieves a high peak signal to noise ratio (PSNR). It means the T-UNet model can effectively upsample the sparse point cloud of low-cost LiDAR to a dense point cloud which is almost indistinguishable from the high-end LiDAR point cloud. The source code can be accessed at https://github.com/donkeyofking/lidar-sr.git

Keywords: LiDAR · Point cloud upsampling · Super-resolution · Temporal convolution · U-Net

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 697–712, 2021.
https://doi.org/10.1007/978-3-030-92635-9_40

1 Introduction

LiDAR (Light Detection and Ranging) is one of the most fundamental and crucial sensors for intelligent robots, self-driving vehicles, and unmanned aerial vehicles (UAV), for it can obtain accurate distance information of the surrounding environments [10,11]. Moreover, LiDAR adopts an active sensing mode that is not affected by ambient light so that it can work at night and dark underground scenes (e.g., caves, mines, and tunnels). Based on the relatively accurate environmental information obtained by LiDAR, sundry robot perception tasks (e.g., target detection and tracking [14,22], segmentation [3,6], simultaneous localization and mapping [13,21,23,31], and navigation [4]) can be performed.

Currently, the off-the-shelf LiDAR can be divided into mechanical LiDAR, and solid-state LiDAR [1]. During the past few decades, we widely use the former in different applications and scenes, and it is designed with 360°of the horizontal field of view (FOV). Furthermore, its vertical field of view is determined by the number of included laser beams and the angle (uniform or non-uniform) between adjacent laser beams (also called angular resolution). The design allows mechanical LiDAR to obtain all the surrounding information within a single scan quickly. However, the predefined angular resolutions (in both horizontal and vertical directions) limit the resolution of perception, especially the horizontal angular resolution. Besides, as the distance increases, the spacing between each laser beam gradually increases. It also results in sparse features in the final 3D point cloud. The solid-state LiDAR fixes this problem with non-repetitive scanning technology by concentrating all the laser beams in a limited field of view, which can scan densely during a certain time interval [2]. Nevertheless, this design sacrifices the field of view range. Furthermore, the complex movement of the mechanical structure to achieve non-repetitive scanning also reduces the final ranging accuracy of the sensor, which is why the FOV of solid-state LiDAR is usually relatively limited. To achieve 360-°horizontal sensing, multiple LiDAR, sophisticated installation structures and synchronization algorithms are usually required [8,18,19].

To avoid the defects involving mechanical LiDAR, the researchers consider the recent progress in point cloud [9,30] and image-based [25] super-resolution technology with deep learning-based modeling. A novel TC [15,16] (Temporal Convolutional) based U-Net model for the point cloud of mechanical LiDAR is proposed. With our model, the sparse point cloud captured by low-cost LiDAR (i.e., Ouster-16/Robosense-32) can be enhanced in real-time and lightweight-edge computing equipment. Moreover, the point cloud upsampled can achieve similar performance compared with the point cloud of high-cost LiDAR (i.e., Ouster-64/Robosense-128). Our T-UNet model can easily extend sparse point cloud from low-cost LiDAR sensor to dense point cloud with dense laser beams, improving the performance of subsequent point cloud-based perception modules. The proposed model can also be treated as a pre-processing module that can be easily inserted into the current workflow of single LiDAR perceptual system or multi-sensor fusion system.

Our research converts the 3D point cloud densification problem into a 2D plane image super-resolution one. The TC module performs like a memory mechanism and tries to capture the inter-frame information to densify and interpolate sparse point clouds. We first project the raw 3D point cloud from the LiDAR with fewer laser beams on a unique 2D plane and enhance the 2D range image with a super-resolution model–T-UNet. Considering the projected point cloud 2D image is much sparser than the camera images and lacks texture information, the enhancement mainly concentrates on the spatial information. As the LiDAR scanning is continuous in the time domain, we extend the typical convolution network into a temporal convolutional one with shared weights. This operation helps fuse information of the time domain while ensuring the lightweight of the model. The structural similarity index measure (SSIM) is introduced as the loss function and ensures that the model can maintain the spatial consistency of the enhanced point cloud. Two scales of enhancement are considered, i.e., upsampling the 16 laser beams into 64 laser beams and upsampling 32 laser beams into 128 laser beams. The final results show that our T-UNet model can achieve a high peak signal to noise ratio (PSNR) on both synthetic data and actual sensor data.

The contributions of this paper are summarized as follows:

- A real-world dataset for 32-to-128 laser beams LiDAR point cloud super-resolution task is released.
- By combining temporal convolutional with U-Net, the proposed T-UNet model can deal with continuous frames and capture the spatiotemporal patterns.
- The dilation convolution is used to replace the pooling layer, and it helps to improve the receptive field without losing information.

The rest of this paper is organized as follows. Sect. 2 briefly reviews the early works on deep learning-based super-resolution tasks and temporal convolutional networks. The detailed design of the T-UNet model is presented in Sect. 3 and the model is evaluated on both the synthetic data captured in the CARLA simulator [7] and the real-world data collected of a local vehicle platform in Sect. 4. Section 5 summarizes the current work and several possible improvements for future research.

2 Related Works

This section introduces the geometric heuristic-based, and deep learning-based progresses on both point cloud and image upsampling or super-resolution tasks. Early research on point cloud upsampling or super-resolution mainly concentrates on point clouds calculated by structured light and a stereo camera. This kind of sensor can only offer approximate 3D distances covering a very close range (i.e., $\leq 10\,\mathrm{m}$) and capture a dense point cloud with both distance and color information. Weinmann et al. [27] refined the resolution limitations of the individual projector-based structured light system with multiple cameras and

projectors and used the iterated bundle adjustment registration of the point cloud from different sensors. This framework aims at reconstructing the whole shape of an object with limited size in an indoor environment. By mining the local triangles relationship built from low-resolution point cloud data, Dinesh et al. [5] proposed a novel bipartite graph approximation-based method with the piecewise-smooth to refine the up-sampled point cloud. Their work can preserve the piecewise smoothness of an object's surface after increasing the point cloud density. Without using additional ancillary data, such as RGB (Red, Green, Blue) color, multiple aligned depth maps, or a database of high-resolution depth exemplars, Michael et al. [12] introduced a depth super-resolution method with the reasoning in terms of patches of 3D points, such as repetition of geometric primitives or object symmetry. With the motion information of the 6-DoF (Degree of Freedom) rigid body, this method can achieve super-resolution of a depth map.

Due to the nonlinear fitting ability and data-driven automatic convergence of the deep learning-based model, recent works try to solve the 3D point cloud super-resolution task with neural networks. For overcoming the limitations of deep learning-based 3D objects super-resolution, the 3D appearance SR (3DASR) dataset [12] is published. The researchers then extended the 2D learning-based SR methods into 3D multi-view tasks by utilizing both the coordinates in 3D space and the texture of color at different layers of the two sub-neural networks. Their work projects the 3D object into a 2D texture map and a normal map, generates a super-resolution texture map, projects it back to 3D space. An adversarial residual graph network [28] is proposed to learn the local similarity and the analogy between low-resolution input and high-resolution output. The residual graph convolution, skip connection design, and a novel loss that combines Chamfer distance and graph adversarial loss extend the normal GCN (Graph Convolution Network). PU-Net [30] tries to learn multi-level features with a multi-branch convolution unit with a joint loss function (i.e., reconstruction loss and repulsion loss). PU-GAN [17] further extends the PU-Net with a generative adversarial network (GAN) and self-attention mechanism. The additional discriminator module helps the generator converge faster. The designers also test PU-GAN on outdoor dataset KITTI [8], but with no ground truth. However, the model parameters and calculations also increase for the additional module. Previous works are mainly designed to generate a high-resolution point cloud of a single object for 3D reconstruction. The super-resolution task for outdoor scenes is rarely considered because LiDAR point clouds are long distances and have too much noise in the outdoor environments.

Shan et al. [24] first projected the 3D point cloud into a 2D image and used an image super-resolution U-Net model to generate a high-resolution image, and back-projected the image into 3D space. The Monte-Carlo dropout is combined to remove noisy points learned by the model. They evaluated their model on a dataset generated from a self-driving simulator. The fundamental problem for designing the outdoor point cloud super-resolution model for self-driving applications with data-driven-based deep learning is the shortage of relevant

datasets. Moreover, the contextual information provided by the spatiotemporal constraint of LiDAR equipped on a vehicle has not been fully exploited. Therefore, we extend this work with temporal convolutional network [15,16,29] and a novel loss function that can guide the model to learn the spatial structure of the environment. We do not simply deploy the model with common used RNN (Recurrent Neural Network) or LSTM (Long Short Term Memory) because they are neither flexible nor suitable for the high-dimensional LiDAR point cloud.

3 Model Architecture

One of the key factors restricting the popularization of autonomous driving and mobile intelligent robots is the cost of multi-beam LiDAR. The point cloud from a low-cost LiDAR is usually sparse and has low resolution, making it difficult for human annotators to recognize objects. Considering the methods mentioned earlier are not designed for real-time situations for on-road 3D LiDAR point cloud densification on self-driving vehicles, we introduce the T-UNet model designed for upsampling the point cloud from a low-cost LiDAR. We aim to generate stable and accurate dense point clouds of the road scenes from consecutive sparse point clouds.

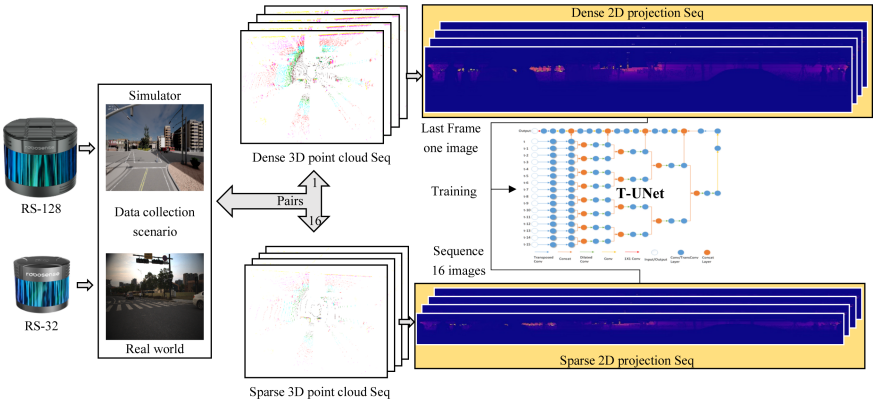


Fig. 1. The proposed T-UNet model for point cloud densification for onboard LiDAR of self-driving vehicle.

Due to the highly correlated consecutive frames from a LiDAR, we take a point cloud sequence instead of one frame pair and learn the inter-frame correlation (temporal and spatial relationships) with a novel TCN framework. To further reduce the computational load of the deep learning model, we transform the 3D space interpolation problem into the super-resolution of the 2D image problem. In this way, we first project the 3D sparse point cloud into a panoramic image according to the coordinating mapping. And then, we encode the feature

maps with transposed convolutional layers with dilated units. Each time we connect the features (only the last frame) with deeper layers (upsampling part) before downsampling the scales of the feature map with a high-way (concatenate) module, helping us combine the TCN with the traditional U-Net architecture. Simultaneously, we combine the feature maps from two adjacent frames to integrate data flows while downsampling. The TCN-based U-Net model tries to recover the blocks with short memories from continuous frames and generates a 2D feature image that can be back-projected to the 3D Cartesian coordinate. This model guarantees the extraction of Spatial-temporal information and hierarchically boosts the processing speed by accelerating continuous point cloud frames in a pipeline. The whole process is illustrated in Fig. 1. We evaluate the present model on CARLA simulator [7] and our platform with a RoboSense Ruby (with 128 laser beams: RS-128).

3.1 Point Cloud Projection and Back-Projection

For processing point cloud of the mechanical LiDAR, uniform or non-uniform laser beams are prefined. The laser angle determines the sensor's vertical field of view (FOV), and its horizontal field of view is 360°. The point cloud density is related to the laser beams (vertical angular resolution) and the rotating speed (horizontal angular resolution). With this information, we can easily project the 3D point cloud on a 2D plane, according to the 3D coordinate (x, y, z), and vertical angle (ω), offset (δ) of azimuth angle (α) according to Eq. 1. When back-projecting the 2D image into 3D spaces according to Eq. 2, we will abandon some points due to the limitation of the predefined image resolution and range distance. In our experiment, about 7% of the points will be lost after the projection and back-projection transformation for each point cloud. However, the lost points take a tiny proportion and can be ignored.

$$\begin{aligned} r &= \sqrt{x^2 + y^2 + z^2} \\ \alpha + \delta &= arctan(y/x) \\ \omega &= arcsin(z/r) \end{aligned} \tag{1}$$

$$\begin{aligned} x &= rcos(\omega)sin(\alpha + \delta) \\ y &= rcos(\omega)cos(\alpha + \delta) \\ z &= rsin(\omega) \end{aligned} \tag{2}$$

According to azimuth and vertical angles, projecting the point cloud is much flexible when dealing with the LiDAR with non-uniform laser beams used in our platform. Figure 2 illustrates the experimental vehicle platform with an RS-32/RS-128 and other equipment. We also illustrate a comparison between Ouster-64 with uniform laser beams and RS-128 with non-uniform laser beams. RS-128's unique design enables the point cloud to be concentrated in the middle of the vertical FOV, thereby helping point clouds obtain more features of on-road targets (such as cars, pedestrians) instead of the ground surface.

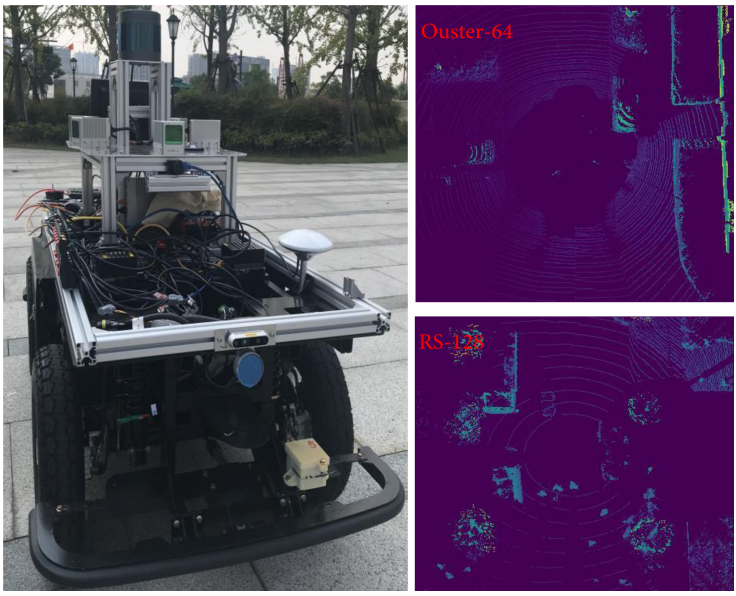
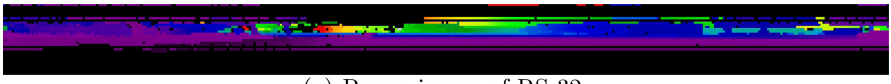
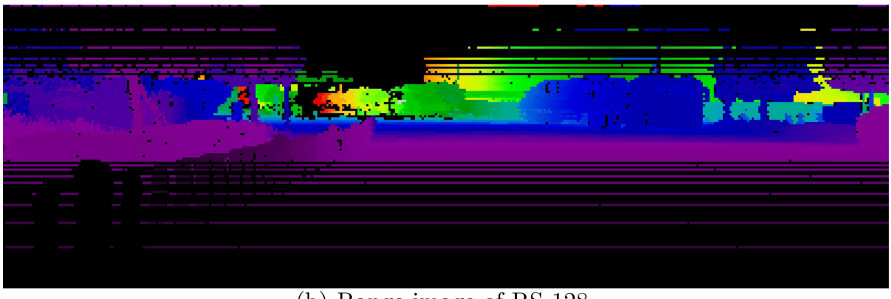


Fig. 2. The vehicle platform with RS-128 (left) and a comparison of bird-view images between RS-128 (non-uniform) and Ouster-64 (uniform).

Figure 3 illustrates the projecting results of the 2D images from RS-32 and RS-128. It can be seen that the widths of the two range images are the same due to the two sensors share the same horizontal FOV and angular resolution. However, the difference is four times the height between the two frames (Fig. 3(a)



(a) Range image of RS-32



(b) Range image of RS-128

Fig. 3. Comparison between range images of RS-32 and RS-128.

and 3(b)). Therefore, our T-UNet model aims to recover a dense range image of RS-128 from sparse range image of RS-32.

We also collected LiDAR data from the CARLA self-driving simulator for model training. However, we only defined uniform LiDAR of Ouster-16 and Ouster-64 with the same vertical FOV but different angle resolutions. Table 1 lists the detailed sensor parameters related to this study.

Table 1. LiDAR parameters used in this study.

Names	Laser beams	FOV (vertical)	Uniform
CARLA16	16	(−15.8°,15.8°)	Yes
CARLA64	64	(−15.8°,15.8°)	Yes
Ouster-16	16	(−15.8°,15.8°)	Yes
Ouster-64	64	(−15.8°,15.8°)	Yes
CARLA32	32	(−25°,15°)	No
CARLA128	128	(−25°,15°)	No
RS-32	32	(−25°,15°)	No
RS-128	128	(−25°,15°)	No

3.2 T-UNet Model

To further exploit the temporal association between consecutive point cloud frames, we combine the temporal convolutional network with U-Net architecture Fig. 4.

U-Net is a classic image segmentation network, which is characterized by U-Shape encoder-decoder structure and skip-connection. We use TCN to modify the encoder module by extracting features from a sequence of low-resolution images and merging them into one feature map. This feature map is equivalent to the feature map of the high-resolution image corresponding to the sequence of low-resolution images. Then the decoder module up samples the feature map and generates the high-resolution image as output.

$$I_t^{low} = \mathcal{P}(PC_t^{low}) \tag{3}$$

$$I_t^{high} = \mathcal{P}(PC_t^{high}) \tag{4}$$

Then we define the object of our model as follows:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}_{SSIM}(\hat{I}_t^{high}, I_t^{high}) + \lambda\Phi(\theta) \tag{5}$$

$$\hat{I}_t^{high} = \mathcal{F}(I_t^{low}, I_{t-1}^{low}, I_{t-2}^{low} \ldots I_{t-(l-1)}^{low}, \theta) \tag{6}$$

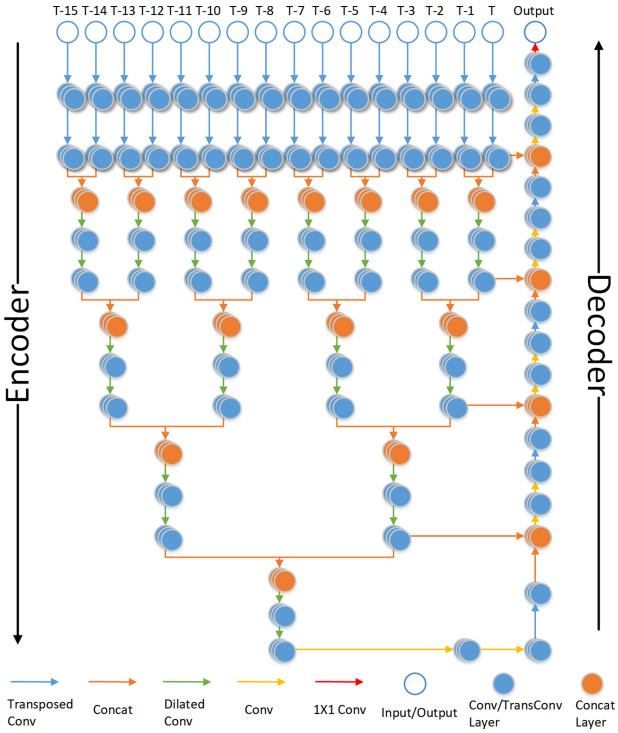


Fig. 4. The architecture of T-UNet model.

where l is the length of low-resolution point cloud sequence, $\mathcal{F}$ defines the function of our model and θ is the parameters of the model. $\mathcal{L}$ represents the SSIM loss function between the generated high-resolution image $\hat{I}_t^{high}$ and the ground truth image I_t^{high}, $\Phi(\theta)$ is the regularization term and λ is the trade-off factor.

4 Experimental Study

4.1 Datasets

Considering the number of existing point cloud super-resolution datasets for the open scene rare, we prepare two datasets from the CARLA simulator and one dataset from the real world. In our experiments, the model up samples I^{low} and generates I^{high}. Therefore, we separately generate I^{low} and I^{high} data with different pre-defined LiDAR in the simulator and on-board LiDAR. Two different upsampling scales are considered, i.e., 16–64 and 32–128; the detailed information is listed in Table 2. The Ouster16-64 is a public released from [24], the CARLA16-64 and CARLA32-128 are synthetic data generated from CARLA, and the RS32-128 is the dataset collected based on our local platform, respectively. For convenience, all datasets are stored in a uniform format.

Table 2. Point cloud super-resolution datasets

Names	Samples	Secne type	FOV (vertical)	Uniform
Carla16-64	6791	Synthetic	(−15.8°,15.8°)	Yes
Ouster16-64	8825	Real-World	(−15.8°,15.8°)	Yes
Carla32-128	6863	Synthetic	(−25°,15°)	No
RS32-128	946	Real-World	(−25°,15°)	No

4.2 Implementation Details

Now we describe the experiments and analyze the performance of the proposed model. We perform 4× upsampling (16 to 64, 32 to 128) for all experiments in this section. Due to the memory limit, we set the batch size to 1 in the training phase. We use $L2$ as the regularization term and set λ in Eq. 5 0.01.

4.3 Model Evaluation

We first compare the present model under different loss functions on the synthetic dataset of Carla16-64. Besides the SSIM, we choose MSE (Mean Square Error) and MAE (Mean Absolute Error) as the loss function for our model. Each model is trained for 20 epochs, and we calculated the PSNR of the ground truth and the generated results from our model as Eq. 7. The PSNR is extended from MSE (Mean Square Error) and describes the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The higher PSNR, the better consistency between the generated data and ground truth.

$$\begin{aligned} PSNR &= 10 \cdot \log_{10}(\frac{MAX_I^2}{MSE}) \\ SSIM &= [l(I_1, I_2)]^{\alpha}[c(I_1, I_2)]^{\beta}[s(I_1, I_2)]^{\gamma} \\ MSE &= \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I_1(i,j) - I_2(i,j)]^2 \end{aligned} \tag{7}$$

where MAX_I is the maximum pixel value of the image, $I_1(i,j)$, $I_2(i,j)$ are two images at the same size. Considering the project images are stored using 8 bits, $MAX_I = 255$. The m and n are the width and height of the projected image. The details about SSIM can be found in [26].

Figure 5 illustrates the PSNRs of the different models during training epochs. It can be seen that the model quickly converged after about six epochs, and the value of PSNR periodically jitter because we use sequence data to train the T-UNet model. However, all the losses of models were finally reduced to an acceptable value under 0.05. Among the three models, the one with the SSIM loss function achieves the highest PSNR from the beginning and can guarantee

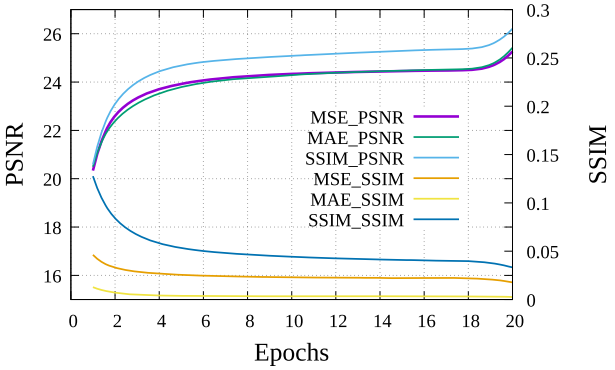


Fig. 5. The PSNR of the models with different loss functions during the training phase.

continued advantages along the whole sequence. Therefore, we use the SSIM as the loss function for the rest evaluation.

We further compare the U-Net model [24] with three different extended versions on both Carla16-64 and Carla32-128. The comparison result also helps us measure how the models deal with data under different scales. Usually the number of laser beams determines the size of a point cloud. Ideally, a LiDAR with 32 laser beams is about twice the point number than 16 laser beams. However, when a laser beam disappears without returning, there is an inevitable fluctuation in the number of points.

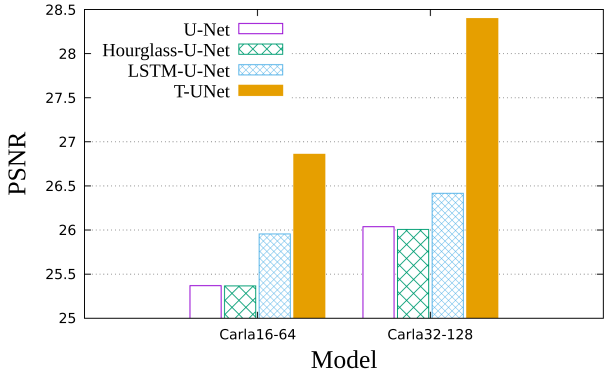


Fig. 6. The PSNR of the four selected models on Carla16-64 and Carla32-128.

Figure 6 illustrates the basic U-Net model and three improved versions. The basic U-Net model only contains an encoder and decoder module with several highway connections between deep (decoder) and shallow (encoder) layers. It is easy to see that when replacing the unstructured U-Net with three stacked

Hourglass modules as backbone [20], the final PSNR has no significant change. However, the model weight scale is only about 1/2 of the original U-Net. We further added an LSTM module at the waist part of U-Net and extracted the most miniature feature map's temporal feature. Furthermore, the final PSNR rises from 25.4 to 25.9. This means the temporal features from continuous frames can effectively help improve the missing point cloud. However, the LSTM module heavily raises the model inference time when adopted on large feature maps. In our case, the LSTM module is deployed on a small feature map, and the promotion is also minimal. Moreover, we combine the TC with the U-Net model instead of only the waist layer in the middle, and the final PSNR rises to 26.83. The T-UNet model achieves the best performance among the modified versions.

When dealing with the Carla32-128 dataset, these models show a very similar distribution as on Carla16-64 but with a higher PSNR value because when dealing with the dense point cloud, the observed information is sufficient for recovering the environment with our model.

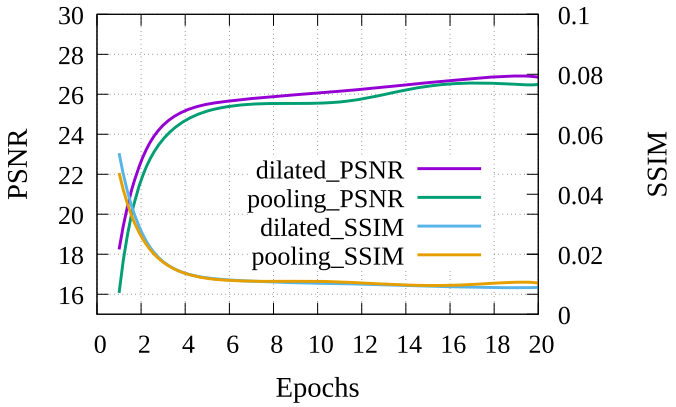


Fig. 7. The T-UNet with pooling layers vs. T-UNet with dilated convolutional layers.

Considering the previous hourglass network can achieve similar performance with fewer model weights, one possible reason is that the max-pooling layers in U-Net may drop out too much information. Therefore, all pooling layers are removed from T-UNet (*pooling_* in Fig. 7), and replaced with the dilated convolutional layers with stride 2 ((*dilated_* in Fig. 7)). The dilated convolution can enlarge the receptive field (kernel size) without enlarging model weights. It can be seen that the T-UNet model with dilated convolution layer achieves more stable PSNR during training epochs than the T-UNet model with max-pooling. Moreover, both models achieve very similar losses on SSIM.

Finally, we test our model on the four datasets mentioned above (two synthetic data and two real-world sensor data) in Fig. 8. It shows that when dealing with a more dense point cloud, the final up-sampled result shows higher PSNR, which means the more information the model gets, the better the final point

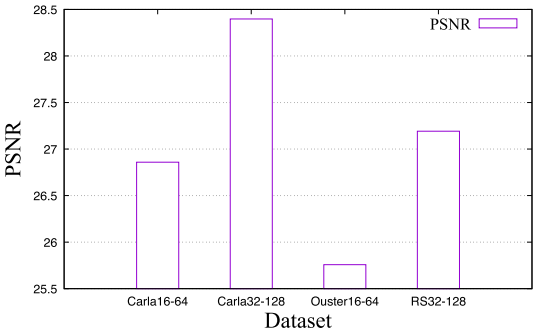
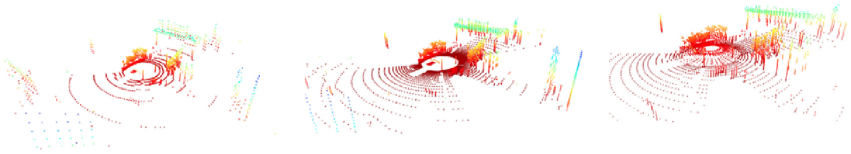


Fig. 8. The performances of the dilated T-UNet models when dealing with different datasets.

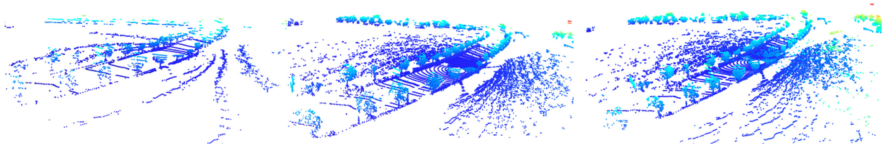
cloud can be restored. The PSNR can achieve nearly 28.5 when using a point cloud of 32 laser beams to generate a point cloud of LiDAR with 128 laser beams. Simultaneously, when dealing with the actual data with more vital randomness, both results reduce the overall PSNR (Ouster16-64 and RS32-128).

Table 3. Time consumptions of the final model on different point scales (on Nvidia GTX2080Ti)

Mode	Processing Time			
	Carla16-64	Ouster16-64	Carla32-128	RS32-128
Single frame	48.6 ms	50.1 ms	152.3 ms	161.2 ms
1000 frames	51.59 s	53.17 s	160.07 s	164.02 s



(a) Ouster16-64: Input (left), GT (middle) and model output(right)



(b) RS32-128: Input (left), GT (middle) and model output(right)

Fig. 9. The super-resolution results. The above pictures are the Ouster16-64 experiment. The below are the RS32-128 experiment. Left: Low-resolution point cloud; Middle: Ground truth; Right: Our model generated.

From Fig. 9, it can be seen that the upsampled point cloud (right) has some noise points compared with ground truth (middle), but it still generates lots of effective points than the original point cloud (left), which can provide more abundant information.

Considering the input point cloud influences the model, we estimated the calculation times of our final dilated T-UNet model on the four different scales of datasets. As listed in Table 3, it can be seen that the model can achieve near 53.0 ms/frame for small scales of a point cloud and 164.0 ms/frame for large scales of the point cloud (point clouds number is about 2 million for a LiDAR with 128 laser beams). Although the current model can not 10 Hz (a typical requirement for autonomous vehicles) when dealing with large-scale data, we think it still has room for improvement.

5 Conclusion

Dense point cloud from high-cost LiDAR supports stable perception for autonomous vehicles and robots. However, the high cost also severely restricts related applications. To achieve better perception performance based on sparse low-end LiDAR, we design a novel TC-based U-Net model for point cloud super-resolution. We first project the 3D point cloud into a 2D image plane, process the continuous frames with dilated convolutional layers, encode the temporal feature with a TC module, and generate super-resolution point feature maps. By re-projecting it back to the 3D space, the dense point cloud is achieved. With the SSIM loss function, the model can capture a more stable spatial consistency of the point cloud. Furthermore, dilated convolutional also helps enlarge the reception field of each kernel without reducing details. The final model can achieve higher PSNR through the improvements as mentioned earlier. In the future, we plan to compress the model with knowledge distillation for better deployment on edge computing devices and real-time processing. The generated result is also considered to be quantitatively evaluated through point cloud-based detection or tracking models.

Acknowledgment. This work has been supported by China Postdoctoral Science Foundation (2020M 681798), Qianjiang Excellent Post-Doctoral Program (2020Y4A001) and 2020 Zhejiang Postdoctoral Research Project (ZJ2020011). JITRI Suzhou Automotive Research Institute Project (CEC20190404). The authors would like to thank Plusgo for their cooperation during data collection.

References

1. Aijazi, A., Malaterre, L., Trassoudaine, L., Checchin, P.: Systematic evaluation and characterization of 3d solid state lidar sensors for autonomous ground vehicles. Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci. **43**, 199–203 (2020)
2. Atanacio-Jiménez, G., et al.: Lidar velodyne hdl-64e calibration using pattern planes. Int. J. Adv. Robot. Syst. **8**(5), 59 (2011)

3. Behley, J., et al.: Semantickitti: a dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 9297–9307 (2019)
4. Christian, J.A., Cryan, S.: A survey of lidar technology and its use in spacecraft relative navigation. In: AIAA Guidance, Navigation, and Control (GNC) Conference, p. 4641 (2013)
5. Dinesh, C., Cheung, G., Bajić, I.V.: 3d point cloud super-resolution via graph total variation on surface normals. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 4390–4394. IEEE (2019)
6. Dong, X., Niu, J., Cui, J., Fu, Z., Ouyang, Z.: Fast segmentation-based object tracking model for autonomous vehicles. In: Qiu, M. (ed.) ICA3PP 2020. LNCS, vol. 12453, pp. 259–273. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60239-0_18
7. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: an open urban driving simulator. arXiv preprint arXiv:1711.03938 (2017)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE (2012)
9. Gevrekci, M., Pakin, K.: Depth map super resolution. In: 2011 18th IEEE International Conference on Image Processing, pp. 3449–3452. IEEE (2011)
10. Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M.: Deep learning for 3d point clouds: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(12), 4338–4364 (2020)
11. Hanson, W., Jones, R., Jones, R.: The roman military presence at dalswinton, dumfriesshire: a reassessment of the evidence from aerial, geophysical and lidar survey. Britannia **50**, 285–320 (2019)
12. Hornacek, M., Rhemann, C., Gelautz, M., Rother, C.: Depth super resolution by rigid body self-similarity in 3d. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1123–1130 (2013)
13. Huang, B., Zhao, J., Liu, J.: A survey of simultaneous localization and mapping. arXiv preprint arXiv:1909.05214 (2019)
14. Huang, R., et al.: An lstm approach to temporal 3d object detection in lidar point clouds. arXiv preprint arXiv:2007.12392 (2020)
15. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156–165 (2017)
16. Lea, C., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks: a unified approach to action segmentation. In: Hua, G., Jégou, H. (eds.) ECCV 2016. LNCS, vol. 9915, pp. 47–54. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49409-8_7
17. Li, R., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-gan: a point cloud upsampling adversarial network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7203–7212 (2019)
18. Lin, J., Liu, X., Zhang, F.: A decentralized framework for simultaneous calibration, localization and mapping with multiple lidars. arXiv preprint arXiv:2007.01483 (2020)
19. Milella, A., Reina, G., Nielsen, M.: A multi-sensor robotic platform for ground mapping and estimation beyond the visible spectrum. Precision Agric. **20**(2), 423–444 (2018). https://doi.org/10.1007/s11119-018-9605-2

20. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29
21. Ouyang, Z., Liu, Y., Zhang, C., Niu, J.: A cgans-based scene reconstruction model using lidar point cloud. In: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 1107–1114. IEEE (2017)
22. Ouyang, Z., Wang, C., Liu, Yu., Niu, J.: Multiview CNN model for sensor fusion based vehicle detection. In: Hong, R., Cheng, W.-H., Yamasaki, T., Wang, M., Ngo, C.-W. (eds.) PCM 2018. LNCS, vol. 11166, pp. 459–470. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00764-5_42
23. Shan, T., Englot, B.: Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4758–4765. IEEE (2018)
24. Shan, T., Wang, J., Chen, F., Szenher, P., Englot, B.: Simulation-based lidar super-resolution for ground vehicles. arXiv preprint arXiv:2004.05242 (2020)
25. Wang, Z., Chen, J., Hoi, S.C.: Deep learning for image super-resolution: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3365–3387 (2020)
26. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). https://doi.org/10.1109/TIP.2003.819861
27. Weinmann, M., Schwartz, C., Ruiters, R., Klein, R.: A multi-camera, multi-projector super-resolution framework for structured light. In: 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, pp. 397–404. IEEE (2011)
28. Wu, H., Zhang, J., Huang, K.: Point cloud super resolution with adversarial residual graph networks. arXiv preprint arXiv:1908.02111 (2019)
29. Yan, J., Mu, L., Wang, L., Ranjan, R., Zomaya, A.Y.: Temporal convolutional networks for the advance prediction of ENSO. Sci. Rep. **10**(1), 1–15 (2020)
30. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-net: Point cloud upsampling network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2790–2799 (2018)
31. Zhang, J., Singh, S.: Loam: lidar odometry and mapping in real-time. In: Robotics: Science and Systems. vol. 2 (2014)